Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y Tecnología

Máster Universitario en Análisis y Visualización de Datos Masivos/ Visual Analytics and Big Data

Modelo de predicción de la demanda para optimizar la producción en planta industrial.

Trabajo fin de estudio presentado por:	Francisco Javier Melero Alegría
Tipo de trabajo:	Desarrollo de Software (Tipo 2)
Director/a:	Ricardo Andrés Fonseca Perdomo
Fecha:	Febrero 2025

Resumen

Este proyecto aborda el desafío de optimizar la planificación de la capacidad productiva en la industria manufacturera, centrándose en una planta del sector del mueble. El propósito principal ha sido el de desarrollar un sistema automatizado capaz de predecir la carga de trabajo en cada línea de producción de la fábrica, permitiendo una gestión eficiente de los recursos humanos y materiales, así como la reducción de ineficiencias como sobrecostes, capacidad ociosa o roturas de stock.

La metodología seguida incluye la extracción y procesamiento de datos históricos de escaneos de productos desde el ERP de la empresa (SAP), para calcular la carga de la fábrica en horas de trabajo semanales y darle forma de serie temporal. Estos datos se han utilizado para entrenar y evaluar cinco modelos predictivos: Suavizado Exponencial, SARIMA, Prophet, XGBoost y redes neuronales LSTM. Cada modelo se ha ajustado para obtener un rendimiento óptimo y ha sido evaluado con métricas de error como RMSE y MAE. Los modelos SARIMA y Suavizado Exponencial han destacado por su precisión, los modelos Prophet y XG Boost han tenido un rendimiento adecuado mientras que el modelo LSTM ha presentado limitaciones debido a la cantidad de datos disponibles.

Los resultados obtenidos indican que la herramienta desarrollada puede predecir la carga de trabajo con un margen de error del 15% al 25% a 16 semanas vista, permitiendo a la planta anticiparse a picos y valles de producción. Además, se ha automatizado el sistema para que las predicciones sean accesibles a través de una interfaz gráfica desarrollada con Streamlit, que permite visualizar y comparar los resultados de los modelos en tiempo real.

En conclusión, este trabajo demuestra la viabilidad de aplicar modelos predictivos avanzados en entornos manufactureros reales, mejorando significativamente la planificación operativa. La herramienta desarrollada representa un paso hacia la digitalización industrial, proporcionando una solución adaptable a otros sectores de la industria manufacturera con características similares.

Palabras clave: Series Temporales, Predicción, Big Data, Industria 4.0.

Abstract

This project addresses the challenge of optimizing production capacity planning in the manufacturing industry, focusing on a furniture manufacturing plant. The main purpose has been to develop an automated system capable of predicting the workload for each production line in the factory, enabling efficient management of human and material resources while reducing inefficiencies such as overcosts, idle capacity, or stock shortages.

The methodology has involved extracting and processing historical product scan data from the company's ERP (SAP) to calculate the workload of the factory measured in weekly working hours, formatted as time series. These data have been used to train and evaluate five predictive models: Exponential Smoothing, SARIMA, Prophet, XGBoost, and LSTM neural networks. Each model has been optimized for maximum performance and has been evaluated using error metrics such as RMSE and MAE. SARIMA and Exponential Smoothing models have stood out for their accuracy, Prophet and XGBoost have delivered adequate performance, while the LSTM model has faced limitations due to the amount of available data.

The results indicate that the developed tool can predict the workload with an error margin of 15% to 25% over a 16-week horizon, allowing the plant to anticipate production peaks and valleys. Furthermore, the system has been automated to make predictions accessible through a graphical interface developed with Streamlit, enabling real-time visualization and comparison of the model´s outputs.

In conclusion, this work demonstrates the feasibility of applying advanced predictive models in real manufacturing environments, significantly improving operational planning. The developed tool represents a step forward in industrial digitalization, offering an adaptable solution for other manufacturing sectors with similar characteristics.

Keywords: Time Series, Forecasting, Big Data, Industry 4.0.

Índice de contenidos

1. Introducción 1
1.1. Motivación 1
1.2. Planteamiento del trabajo 2
1.3. Estructura del trabajo 3
2. Contexto y estado del arte 5
2.1. Contexto del problema 5
2.2. Estado del arte 7
2.3. Marco teórico 12
2.3.1. Exponential Smoothing. 12
2.3.2. SARIMA 14
2.3.3. Prophet 17
2.3.4. XGBoost 20
2.3.5. Redes neuronales LSTM 23
2.4. Conclusiones 24
3. Objetivos concretos y metodología de trabajo 25
3.1. Objetivo general 25
3.2. Objetivos específicos 25
3.3. Metodología del trabajo 25
4. Marco normativo 29
5. Desarrollo específico de la contribución 31
5.1. Requisitos técnicos 31
5.2. Tecnologías empleadas 32
5.3. Tratamiento de datos y EDA 34
5.3.1. Dataset de partida. 34

5.3.2. Limpieza de datos 36
5.3.3. Análisis exploratorio de datos (EDA). 38
5.3.4. Medición de la cantidad de trabajo. 41
5.3.5. Estacionalidad, y tendencia de la serie temporal 47
5.4. Entrenamiento y evaluación de los modelos 49
5.4.1. Exponential Smoothing – Holt Winters 51
5.4.2. SARIMA 55
5.4.3. Prophet. 60
5.4.4. XGBoost 64
5.4.5. Red Neuronal LSTM 69
5.5. Comparación de los modelos. 73
5.6. Automatización de la predicción. 78
5.6.1. Backend 78
5.6.2. Frontend 78
6. Código fuente y datos analizados 83
6.1. Código fuente 83
6.2. Datos Analizados 83
7. Conclusiones 84
8. Limitaciones y prospectiva 86
8.1. Limitaciones 86
8.2. Trabajo futuro 87
Referencias bibliográficas 89

Índice de Figuras

Figura 1: *Línea de fabricación de volúmenes (Armarios, taquillas, archivos…)*.....5

Figura 2: *Línea de fabricación de patas de mesa* 6

Figura 3: *Línea de fabricación de tableros de mesa* 6

Figura 4: *Línea de fabricación de paneles separadores*.....6

Figura 5: *Predicción errónea del número de eventos en Facebook con distintos algoritmos.* 18

Figura 6: *Efecto de distintos L y Ω al ajustar un modelo de escalón.* 21

Figura 7: *Estructura de una celda LSTM. Fuente: Mathworks*.....23

Figura 8: *Fases del proyecto* 26

Figura 9: *Proceso de escaneo de la orden de trabajo del producto final.* 34

Figura 10: *Productos fabricados en cada línea de fabricación* 38

Figura 11: *Evolución de productos fabricados. Años 2021-2024*.....39

Figura 12: *Productos y cantidades fabricadas en la línea de volúmenes. Años 2021-2024* 40

Figura 13: *Línea de producción y Takt Time* 42

Figura 14: *Histograma de Takt Times en la línea de volúmenes.* 44

Figura 15: *Serie temporal de carga de trabajo para cada día en línea de volúmenes.* 45

Figura 16: *Serie temporal de carga de trabajo para cada semana en línea de volúmenes.* ...46

Figura 17: *Serie temporal de cantidad de productos para cada semana en línea de volúmenes.* 46

Figura 18: *Tendencia, Estacionalidad y Residuo de la serie temporal de carga de trabajo semanal.* 47

Figura 19: *División de datos de entrenamiento y test.* 49

Figura 20: *Predicción de los datos de test en Holt-Winters con ajuste iterativo de hiperparámetros* 51

Figura 21: *Error en la predicción de los datos de test en Holt-Winters con ajuste iterativo de hiperparámetros* 52

Figura 22: *Predicción errónea de valores futuros usando Holt-Winters con ajuste iterativo de hiperparámetros* 52

Figura 23: *Predicción de los datos de test en Holt-Winters con ajuste automático MLE de hiperparámetros* 53

Figura 24: *Error en la predicción de los datos de test en Holt-Winters con ajuste automático MLE de hiperparámetros* 53

Figura 25: *Predicción de la carga de planta en las próximas 16 semanas con el modelo de Holt-Winters* 54

Figura 26: *Predicción de los datos de test en SARIMA con ajuste iterativo de hiperparámetros* 56

Figura 27: *Error en la predicción de los datos de test en SARIMA con ajuste iterativo de hiperparámetros* 56

Figura 28: *Predicción errónea de valores futuros usando SARIMA con ajuste iterativo de hiperparámetros* 57

Figura 29: *Búsqueda en el grid de hiperparámetros p-q del mejor ajuste de SARIMA.* 58

Figura 30: *Predicción de los datos de test en SARIMA con ajuste automático de hiperparámetros* 58

Figura 31: *Error en la predicción de los datos de test en SARIMA con ajuste automático de hiperparámetros* 59

Figura 32: *Predicción de la carga de planta en las próximas 16 semanas con el modelo SARIMA* 59

Figura 33: *Tendencia de la serie temporal según Prophet con "changepoint_prior_scale" optimizado iterativamente* 61

Figura 34: *Estacionalidad de la serie temporal según Prophet* 61

Figura 35: *Efectos de los cierres según Prophet.* 61

Figura 36: *Predicción de los datos de test en Prophet con ajuste automático de hiperparámetros* 62

Figura 37: *Error en la predicción de los datos de test en Prophet.* 62

Figura 38: *Predicción de la carga de planta en las próximas 16 semanas con el modelo Prophet.* 63

Figura 39: *Representación de secuenciación de serie temporal para entrenamiento de XGBoost.* 64

Figura 40: *Predicción de los datos de test en XGBoost con ajuste iterativo de hiperparámetros* 66

Figura 41: *Error en la predicción de los datos de test en XGBoost* 66

Figura 42: *Predicción de los datos de test en XGBoost con corrección de error.* 67

Figura 43: *Predicción de la carga de planta en las próximas 16 semanas con el modelo XGBoost.* 68

Figura 44: Pérdida de entrenamiento y validación durante el entrenamiento de la red LSTM. 70

Figura 45: *Predicción de los datos de test en red neuronal LSTM (detalle).* 70

Figura 46: *Error en la predicción de los datos de test en red neuronal LSTM* 71

Figura 47: *Predicción de la carga de planta en las próximas 16 semanas con el modelo red neuronal LSTM.* 71

Figura 48: *Comparación de la predicción final de distintos modelos vs valores reales.* 74

Figura 49: *Comparación de la predicción final de distintos modelos vs valores reales (detalle)* 74

Figura 50: *RMSE de la predicción de la carga de planta a 16 semanas vista.* 75

Figura 51: *MAE de la predicción de la carga de planta a 16 semanas vista.* 75

Figura 52: *Frontend de la aplicación mostrando las predicciones de todos los modelos para la planta completa.* 79

Figura 53: *Frontend de la aplicación mostrando las predicciones del modelo Exponencial para la línea de Paneles.* 80

Figura 54: *Frontend de la aplicación mostrando las predicciones con zoom de todos los modelos para la línea de Tableros.* 81

Figura 55: *Detalle de la gráfica interactiva de exploración de predicciones en modo pantalla completa*. 82

Índice de tablas

Tabla 1: *Hiperparámetros en el suavizado exponencial de Holt-Winters* 13

Tabla 2: *Hiperparámetros para XGBoost.* 22

Tabla 3: *Dataset de escaneos de productos en planta (Dataset de partida)* 35

Tabla 4: *Equivalencia entre línea de fabricación y código de escáner.* 36

Tabla 5: *Datos de partida tras limpieza y pre-tratamiento.* 37

Tabla 6: *Takt Times y desviación estándar para los productos más fabricados en la línea de volúmenes* 43

Tabla 7: *Cantidad de productos, y carga de trabajo para la línea de volúmenes* 45

Tabla 8: *Métricas de error para el modelo de Holt-Winters* 54

Tabla 9: *Mejores hiperparámetros para el ajuste iterativo del modelo SARIMA.* 55

Tabla 10: *Métricas de error para el modelo SARIMA* 59

Tabla 11: *Fechas de cierre programado para la serie.* 60

Tabla 12: *Métricas de error para el modelo Prophet.* 63

Tabla 13: *Datos de entrenamiento para XGBoost.* 65

Tabla 14: *Métricas de error para el modelo XGBoost* 67

Tabla 15: *Métricas de error a 8 y 16 semanas para los modelos.* 76

1. Introducción

1.1. Motivación

En el campo de la ingeniería industrial, uno de los problemas que se presentan con más frecuencia es el de la planificación de las capacidades productivas de una o varias fábricas. Para que una industria sea capaz de producir los bienes demandados por el mercado en términos de calidad y tiempo, no solamente es necesario que los equipos productivos se encuentren en perfecto estado, sino que además es preciso que la planta cuente con el personal y las materias primas necesarios para poder satisfacer las necesidades del cliente.

De este modo, es común que en diversas industrias se realice una planificación de las necesidades para poder satisfacer correctamente la demanda en base a un conocimiento experto adquirido con la experiencia a lo largo de los años, pero que a menudo peca de ser poco riguroso y cuantitativo. Es común oír decir que se espera “alta carga” o “baja carga” en una empresa sin realmente poner cifras a dicha carga de trabajo y sin especificar cuáles son las fluctuaciones esperadas en la demanda. Este problema es más común en industrias que producen una gran diversidad de productos o trabajan por proyectos (como la industria del mueble) que en industrias donde se produce una gran cantidad de productos estándar (como la industria automotriz), ya que en las primeras la carga de trabajo es más difícil de cuantificar de forma objetiva.

Esta falta de concreción a la hora de cuantificar y predecir la carga de una industria produce grandes ineficiencias (o MUDA, empleando terminología Lean Manufacturing) que a su vez derivan en sobrecostes relacionados con la necesidad de trabajar horas extra para producir los bienes necesarios o, por el contrario, contar con capacidad ociosa en planta. Como señala Ohno (1988), la identificación y eliminación de desperdicios, incluyendo aquellos derivados de una planificación deficiente, es clave para mejorar la eficiencia y reducir los costos operativos.

Como consecuencia, el desarrollo de soluciones de software que permitan realizar una predicción de la demanda o carga de trabajo en una industria, es una potente herramienta a la hora de reducir costes en todos los departamentos de la empresa. Según Stevenson (2021), la integración de modelos predictivos basados en datos históricos y el uso de tecnologías avanzadas de análisis permite mejorar significativamente la toma de decisiones en la gestión de la capacidad productiva.

1.2. Planteamiento del trabajo

En el contexto de la industria manufacturera, la planificación eficiente de la capacidad productiva es un problema transversal a todas las empresas. Las fábricas necesitan adaptarse rápidamente a las fluctuaciones en la demanda para evitar ineficiencias como la sobreproducción, el desperdicio de recursos o tiempos de inactividad en las líneas de fabricación. La optimización de los recursos productivos para evitar ineficiencias es un tema tratado ampliamente en la literatura con obras como The Toyota Way Liker (Liker, 2004) o Lean Manufacturing: Fundamentals, Tools, Approaches, and Industry 4.0 (Vinodh, 2021), donde se explora la manera de reducir los stocks, la sobrecapacidad y la infrautilización de equipos en el entorno industrial.

No obstante, la literatura actual en mejora de procesos industriales plantea un enfoque principalmente reactivo al problema, enfocándose en estrategias para adaptarse eficientemente a los picos y valles en la demanda una vez se han producido. Sin embargo, este proyecto propone un enfoque proactivo, basado en la anticipación de la demanda futura mediante el uso de datos históricos almacenados en el ERP de la empresa (SAP) y su análisis mediante modelos predictivos avanzados.

La viabilidad del proyecto se basa en dos factores clave: la disponibilidad de datos históricos confiables y el uso de herramientas estadísticas y algoritmos de aprendizaje automático para realizar predicciones con alta precisión, dado que estos algoritmos ya se emplean con éxito en otros sectores como el financiero, como demuestra Ruey S. Tsay en su obra “Analysis of financial time series” (Tsay, 2020).

A diferencia de los enfoques tradicionales, esta propuesta no busca adaptarse a la demanda fluctuante, sino preverla con suficiente antelación para facilitar una planificación más eficiente de los recursos productivos. Además, la solución incluye una interfaz gráfica que permite que el personal de la empresa pueda interpretar y utilizar los resultados sin necesidad de conocimientos técnicos avanzados, asegurando así la aplicabilidad operativa.

La principal pregunta que guía esta investigación es: ¿cómo pueden los modelos predictivos de series temporales ser integrados en una solución de software que permita, a partir de la cuantificación precisa de la carga de trabajo en planta, anticipar la demanda y facilitar la planificación de los medios productivos?

Este enfoque no solo busca aportar una herramienta tecnológica innovadora, sino también sentar las bases para una gestión más proactiva y eficiente de los recursos en la industria manufacturera.

1.3. Estructura del trabajo

Este proyecto se ha dividido en una serie de capítulos diferenciados. A lo largo de esta sección se describe brevemente los diferentes apartados de los que consta el documento y el contenido de cada uno de ellos.

En la sección 1, dedicada a la introducción, se explica la motivación por la cual se lleva a cabo este proyecto así como el planteamiento de este, pasando de una necesidad genérica que se produce en el mundo de la empresa a un planteamiento concreto para resolver dicho problema.

A lo largo de la sección 2, se contextualiza el problema con más detalle dentro del entorno empresarial. Además, se estudian las distintas soluciones ya existentes para el mismo, identificando sus fortalezas y limitaciones. Se describe también el marco teórico que se empleará para resolver el problema, explicando los distintos algoritmos que se han usado a lo largo del proyecto.

En la sección 3 se parte del planteamiento y el contexto desarrollados en las secciones 1 y 2 para establecer una serie de objetivos, tanto generales como específicos, para llevar a cabo el proyecto y dar solución a los problemas planteados. Por otra parte, se establece una metodología de trabajo clara y unas fases de resolución del problema que se seguirán a lo largo del proyecto.

En la sección 4 se describe el manejo de datos en el TFE, limitado a información operativa no identificable. Se describe la implementación de medidas de seguridad como acceso restringido y almacenamiento seguro, además de documentar el proceso para garantizar trazabilidad y responsabilidad. Aunque no ha sido necesario aplicar el RGPD, se han seguido prácticas éticas en la gestión de datos empresariales.

La sección 5, sección principal del proyecto, detalla la contribución específica realizada. En primer lugar, se describen los requisitos técnicos las tecnologías utilizadas y el proceso de limpieza y análisis exploratorio de datos para, a continuación, cuantificar la cantidad de trabajo en planta. Luego, se evalúan y comparan distintos modelos predictivos de series

temporales. Finalmente, se automatiza la predicción y se desarrolla una interfaz gráfica para explorar los resultados obtenidos.

En la sección 6 se incluye el código fuente del proyecto, alojado en un repositorio de GitHub. Este contiene todo lo necesario para reproducir los resultados obtenidos a lo largo del proyecto. Debido a la confidencialidad de los datos no se proporciona acceso a los mismos, esto garantiza el cumplimiento de las medidas de seguridad y confidencialidad requeridas por la empresa.

En la sección 7 se exponen las conclusiones del proyecto, evaluando el cumplimiento de los objetivos planteados y reflexionando sobre los resultados obtenidos. Se destacan las principales aportaciones realizadas durante el desarrollo del trabajo y la relevancia de estas en el contexto del problema abordado.

Finalmente, en la sección 8 se analizan las limitaciones encontradas durante el desarrollo del proyecto, identificando los aspectos que pudieron restringir su alcance o impacto. Además, se plantean posibles líneas de desarrollo futuro que podrían mejorar las capacidades del sistema, incrementar su precisión o ampliar su aplicabilidad en otros contextos o sectores, explorando oportunidades para evolucionar el trabajo realizado.

2. Contexto y estado del arte

2.1.Contexto del problema

Este trabajo se desarrolla en el seno de una conocida empresa multinacional de mobiliario de oficinas. La empresa posee 4 plantas en Europa, aunque este proyecto se lleva a cabo en la planta de Madrid, que cuenta con unos 300 empleados.

En dicha empresa, la producción se lleva a cabo en 5 líneas de producción principales que trabajan en paralelo fabricando distintos productos. Cada línea de fabricación se encarga de producir productos de una misma familia, siendo estas familias: Volúmenes (armarios, taquillas, archivos…), Patas de mesa, Tableros de mesa, Paneles Separadores y Varios (otros productos).

En la figura 1 puede observarse la línea de fabricación de volúmenes.

Figura 1:

Línea de fabricación de volúmenes (Armarios, taquillas, archivos…)

En la figura 2 se muestra la línea de fabricación de patas de mesa.

Figura 2:

Línea de fabricación de patas de mesa

En la figura 3 se aprecia la línea de fabricación de tableros de mesa.

Figura 3:

Línea de fabricación de tableros de mesa

En la figura 4 se observa la línea de fabricación de paneles separadores.

Figura 4:

Línea de fabricación de paneles separadores

Actualmente en la empresa no existe un modelo formal de predicción de la demanda, por el contrario, se emplea el conocimiento experto y la experiencia en años anteriores para hacer una estimación de la demanda, que se va definiendo conforme van haciéndose pedidos por parte de los diversos clientes.

2.2. Estado del arte

El estudio de la predicción de series temporales es una disciplina ampliamente utilizada en el mundo empresarial, aunque con una predominancia en sectores como el financiero o la consultoría en comparación con el sector industrial (Blanco, García, & Remesal, 2023). Estas técnicas permiten anticipar tendencias y comportamientos futuros a partir de datos históricos, convirtiéndose en herramientas esenciales para la toma de decisiones estratégicas.

Una de las referencias más relevantes en este campo es Forecasting: Principles and Practice (Hyndman & Athanasopoulos, 2021), que aborda la necesidad de proporcionar una guía práctica y accesible para el aprendizaje del pronóstico de series temporales. Los autores destacan la aplicabilidad de modelos clásicos como Exponential Smoothing y SARIMA, mostrando su robustez en escenarios donde la simplicidad y la interpretación son clave. Esta obra es el principal documento de referencia que se ha empleado para el desarrollo de este proyecto.

En una línea similar, Time Series Analysis: Forecasting and Control (Box, Jenkins, Reinsel, & Ljung, 2015) es un referente en la modelización de series temporales con patrones complejos y estacionales. Este libro introduce la metodología Box-Jenkins, un enfoque estructurado que ha sido adoptado ampliamente para ajustar modelos ARIMA y SARIMA en diversas aplicaciones. Su contribución radica en la capacidad de estos modelos para capturar tanto dinámicas estacionales como no estacionales, lo que los convierte en herramientas versátiles para diferentes contextos industriales y económicos.

En investigaciones más recientes, Time Series Clustering Based on Prediction Accuracy of Global Forecasting Models (López Oriona, Montero Manso, & Vilar Fernández, 2023) propone un enfoque innovador para la clasificación de series temporales basado en su predictibilidad. Este trabajo explora cómo los modelos globales pueden optimizar la agrupación de series con características similares, una herramienta valiosa para aplicaciones industriales donde se

gestionan múltiples productos o líneas de fabricación. Este avance refuerza la idea de que las técnicas modernas pueden complementar y potenciar enfoques clásicos.

Otro avance significativo es el modelo FPN-fusion, presentado por Li, Xiao y Yuan (2024). Este trabajo responde a la creciente demanda de soluciones computacionalmente eficientes en el ámbito de la predicción de series temporales. Al implementar técnicas de fusión multinivel y redes de pirámide de características, los autores demuestran que es posible mantener un alto nivel de precisión mientras se reduce la complejidad computacional, lo que resulta crucial en entornos industriales con limitaciones de recursos. Este enfoque es de especial relevancia, ya que hay modelos como SARIMA, que requieren una gran cantidad de tiempo para realizar el ajuste de sus hiperparámetros, especialmente cuando este se realiza de forma automática.

De forma similar a López, Monteo y Vilar, la utilidad de modelos globales para la predicción de series temporales múltiples es explorada por Hewamalage, Bergmeir y Bandara (2020), quienes comparan su desempeño con enfoques univariados tradicionales. Este estudio demuestra que los modelos globales no solo son eficaces en términos de precisión, sino que también ofrecen ventajas significativas en términos de escalabilidad, particularmente en contextos industriales donde se deben gestionar simultáneamente múltiples líneas de datos.

En el ámbito energético, Lago, Marcjasz, De Schutter y Weron (2021) destacan la importancia de combinar algoritmos tradicionales y modernos para la predicción de precios de electricidad. Este trabajo no solo proporciona una revisión exhaustiva de las técnicas existentes, sino que también establece un punto de referencia para futuras investigaciones, subrayando cómo los enfoques híbridos pueden ofrecer resultados superiores.

Asimismo, Orlando, Bufalo y Stoop (2022) aplican modelos matemáticos avanzados para analizar aspectos deterministas en series temporales financieras, proporcionando una perspectiva innovadora sobre la dinámica de estos mercados. Este enfoque, aunque centrado en finanzas, es extrapolable a otros sectores, demostrando que las series temporales pueden descomponerse y modelarse incluso en contextos complejos.

Respecto a la precisión de los diferentes modelos de series temporales existentes, la M4 Competition evaluó 100,000 series temporales de seis dominios (microeconómico, industrial, macroeconómico, financiero, demográfico y otros) con frecuencias desde anual hasta horaria, abarcando diversas aplicaciones como planificación estratégica y operaciones. Comparó

métodos estadísticos, de aprendizaje automático (ML) y combinados, destacando que los enfoques híbridos, como la integración de redes neuronales recurrentes (RNN) y modelos estadísticos, lograron la mayor precisión.

Aunque los métodos puramente enfocados en el ML tuvieron limitaciones, los combinados y los intervalos de predicción precisos marcaron un avance significativo. La competencia subrayó la importancia de combinar enfoques y promover la reproducibilidad, estableciendo un nuevo estándar para la predicción de series temporales (Makridakis, Spiliotis, & Assimakopoulos, 2020). Esto indica que en diversos contextos los modelos de Machine Learning puros no presentan por lo general un rendimiento tan alto como los modelos estadísticos híbridos.

En el contexto industrial, Madrigal Espinosa (2014) y García (2015) presentan enfoques orientados a la mejora de la toma de decisiones estratégicas. Madrigal analiza la eficacia de los modelos de regresión para series con estacionalidad creciente, mientras que García proporciona herramientas prácticas para gestionar datos temporales en entornos empresariales, destacando el valor de los métodos estadísticos tradicionales como punto de partida para optimizar procesos.

Aplicación al dimensionamiento y control de sistemas altamente variables (Crespo Pereira, 2013) presenta un enfoque práctico para la optimización de líneas de fabricación. Este trabajo se centra en la modelización de series temporales para capturar variabilidad y autocorrelación, aplicando estas técnicas en sistemas de producción altamente variables. Los resultados demuestran que este enfoque mejora significativamente el rendimiento de los sistemas industriales y la gestión de recursos.

De igual manera, García (2015) en Predicción en el dominio del tiempo: Análisis de series temporales para ingenieros, ofrece herramientas prácticas para la modelización y predicción de series temporales. En este trabajo se presentan modelos clásicos como ARIMA y suavizado exponencial, mostrando su eficacia en la predicción a corto plazo y su aplicabilidad en la toma de decisiones estratégicas en contextos industriales.

En el contexto de machine learning, el estudio Machine Learning para la predicción de series temporales en indicadores de desarrollo (Banco Mundial, 2021) explora la aplicación de técnicas modernas de aprendizaje automático para predecir indicadores de desarrollo

económico. Este trabajo demuestra cómo modelos basados en árboles de decisión y bosques aleatorios pueden ser efectivos para identificar patrones y relaciones complejas en los datos, con aplicaciones extrapolables al sector industrial.

Por otro lado, Ortega y Santamaría (2020), en su trabajo Análisis de series temporales: Estudio de un caso práctico, implementan modelos ARIMA en el análisis del Euríbor, destacando su capacidad para capturar dinámicas temporales en series financieras. Este enfoque resalta la robustez de los métodos estadísticos clásicos en entornos con alta variabilidad y su potencial aplicabilidad en la industria manufacturera.

Smith (2024), en Large Scale Hierarchical Industrial Demand Time-Series Forecasting Incorporating Sparsity, aborda el problema de la predicción jerárquica de series temporales en aplicaciones industriales, donde los datos de niveles inferiores de la jerarquía presentan una alta esparcidad. La metodología propuesta incluye un modelo probabilístico jerárquico que adapta supuestos distribucionales según la densidad de las series temporales, asegurando la coherencia en la reconciliación jerárquica. Los resultados muestran una mejora en la precisión de las predicciones en un 8.5% para series densas y un 23% para series esparsas, con aplicaciones directas en la planificación empresarial y la experiencia del cliente.

Por su parte, Kovacs (2023) en An Adaptive Learning Approach to Multivariate Time Forecasting in Industrial Processes explora la necesidad de mejorar la predicción multivariante de variables temporales en procesos industriales para optimizar políticas de mantenimiento. La metodología incluye un modelo oculto de Markov integrado con un algoritmo adaptativo que incorpora datos nuevos en tiempo real. Los resultados demostraron una mejora consistente en comparación con enfoques univariantes, destacando su aplicabilidad en sectores industriales que manejan grandes volúmenes de datos interrelacionados.

Liu (2024), en su trabajo Multi-variable Adversarial Time-Series Forecast Model, se centra en la predicción a corto plazo de sistemas eléctricos en empresas industriales. Este problema implica manejar múltiples variables críticas para el control de carga y la protección de maquinaria. La metodología empleada combina redes neuronales de tipo Long Short-Term Memory (LSTM) con un enfoque adversarial para regularizar las predicciones. Los resultados muestran una mejora significativa en la precisión, especialmente en datos provenientes de monitores avanzados de energía, superando enfoques tradicionales.

Wang (2023), en Industrial Forecasting with Exponentially Smoothed Recurrent Neural Networks, plantea el problema de modelar sistemas dinámicos no estacionarios característicos de aplicaciones industriales. La metodología incluye una clase de redes neuronales recurrentes suavizadas exponencialmente (RNNs) que capturan estructuras de autocorrelación no lineales, integrando estacionalidad y tendencias. Los resultados indican que este enfoque supera a modelos como ARIMA y redes neuronales recurrentes simples en la predicción de cargas eléctricas y datos meteorológicos, ofreciendo una alternativa eficiente para problemas complejos.

Finalmente, Chen (2024), en Deep Learning-based Time Series Forecasting: Models and Applications, aborda la predicción de series temporales complejas en entornos industriales. El problema principal es manejar la heterogeneidad, no linealidad e incertidumbre inherentes en los datos. La metodología incluye redes neuronales residuales ruidosas y métodos de ensamblado de instantáneas para la estimación puntual y la cuantificación de incertidumbre. Los resultados destacan una mejora en la precisión y una mayor robustez en la predicción, con aplicaciones en sectores que requieren decisiones críticas basadas en datos complejos.

En conclusión, aunque la teoría y los modelos de predicción de series temporales están ampliamente desarrollados y aplicados en diversos sectores como el financiero, el energético o la predicción de fenómenos naturales, la industria manufacturera aún carece por lo general de soluciones personalizadas y ampliamente adoptadas en el entorno empresarial para la cuantificación y predicción de la carga de trabajo. Este vacío representa una oportunidad única para adaptar y aplicar estas técnicas a contextos específicos, sentando las bases para su futura adopción comercial en el sector industrial.

2.3. Marco teórico

A lo largo de esta sección se describen los fundamentos teóricos de los algoritmos utilizados en este trabajo para la predicción de series temporales. Cada uno de estos algoritmos es empleado a lo largo de la sección 5.4 de manera práctica.

2.3.1. Exponential Smoothing.

El método de suavizado exponencial es ampliamente utilizado para el pronóstico de series temporales debido a su simplicidad y efectividad para capturar tendencias y estacionalidades.

La técnica de Exponential Smoothing (o suavizado exponencial) fue propuesta a lo largo de finales de los años 50 y principios de los años 60 por los ingenieros Charles C. Holt y Peter Winters (Winters, 1960), y es uno de los métodos de pronóstico de series temporales más sencillos y efectivos.

Este enfoque asigna pesos decrecientes a los datos históricos, priorizando las observaciones más recientes. Holt y Winters introdujeron extensiones como la suavización exponencial doble y triple para manejar tendencias y estacionalidades respectivamente (Holt, 2004; Winters, 1960). Su aplicación en entornos industriales ha demostrado ser efectiva para la planificación de inventarios y el control de producción (Chatfield, 2003).

El método en su forma más básica (Suavizado Exponencial Simple), consiste en asumir que cada valor en un momento dado, se puede calcular como una media ponderada de los valores pasados, donde los valores recientes tienen más peso que los anteriores:

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots \quad (1)$$

o, más generalmente:

$$\hat{y}_{T+1|T} = \sum_{j=0}^{T-1} \alpha(1-\alpha)^j y_{T-j} + (1-\alpha)^T \ell_0 \quad (2)$$

Donde 0 <= α <= 1 es el parámetro de suavizado, que controla la velocidad a la que ha de descender el peso que se le da a cada valor pasado conforme las observaciones se alejan en el tiempo. Cuanto más cercano a 1 es α, menos peso se otorga a las observaciones más alejadas en el tiempo y viceversa.

Por otro lado, l0 es el "nivel inicial" de la serie temporal, que agrupa todos los valores anteriores a él de la serie, para que estos no sean descartados del todo en series muy largas. En el caso de series temporales no demasiado largas (como es el caso) l0 puede coincidir con el primer valor de la serie.

Sin embargo, como se muestra en la figura 18, la serie temporal a pronosticar posee estacionalidad y tendencia. Por esta razón el modelo anterior no es suficiente para realizar un pronóstico adecuado, ya que no captura correctamente todas las componentes de la serie temporal.

Existen dos modelos de Holt-Winters para incorporar estacionalidad y tendencia a las predicciones de la serie temporal. A lo largo de este trabajo se ha empleado el modelo aditivo, que se muestra en la fórmula a continuación:

$$\begin{aligned}\hat{y}_{t+h|t} &= \ell_t + hb_t + s_{t+h-m(k+1)} \\ \ell_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \\ s_t &= \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m},\end{aligned} \quad (3)$$

Donde, ℓt es el valor base o promedio en el momento actual, bt es la tendencia o ritmo de cambio a largo plazo y *st* es la estacionalidad o fluctuación periódica, dando lugar por tanto a los hiperparámetros mostrados en la tabla 1:

Tabla 1:

Hiperparámetros en el suavizado exponencial de Holt-Winters

Hiperparámetro	Descripción	Valores
α	Factor de suavizamiento para el **nivel**: controla la rapidez con que el nivel se ajusta.	0-1
β	Factor de suavizamiento para la **tendencia**: controla la rapidez con que la tendencia se ajusta.	0-1
γ	Factor de suavizamiento para la **estacionalidad**: controla la rapidez con que la estacionalidad se ajusta.	0-1
m	Longitud del ciclo estacional: número de periodos en un ciclo estacional completo.	52

2.3.2. SARIMA

El modelo SARIMA (Seasonal AutoRegressive Integrated Moving Average) es uno de los métodos más usados para la predicción de series temporales. Combina componentes autoregresivos, de promedio móvil y de integración, extendiendo el clásico ARIMA para manejar estacionalidad. Este modelo es ampliamente utilizado para series temporales con patrones estacionales regulares, como las ventas mensuales o los datos de producción semanal (Wei, 2019). Además, su robustez para capturar dinámicas complejas lo ha hecho popular en aplicaciones de pronóstico en manufactura y logística (Shumway & Stoffer, 2017).

Mientras que los modelos de suavizado exponencial se basan en la descripción de la tendencia y estacionalidad de la serie, los modelos ARIMA tratan de describir autocorrelaciones dentro de los datos. Es por eso por lo que los modelos ARIMA poseen tres componentes principales que se detallan a continuación.

- AR (AutoRegresivo): El valor de la serie actual es una combinación lineal de sus valores pasados. Por tanto, la componente AR puede ser descrita matemáticamente como:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t, \quad (4)$$

  y_t = valor de la serie en un momento determinado.

  c = Constante. Su cálculo forma parte del entrenamiento del modelo.

  ϕ_n = Coeficiente autorregresivo n, representa el grado de influencia de cada valor pasado. A diferencia del suavizado exponencial, estos valores no tienen por qué decrecer a medida que los valores se alejan en el tiempo. Su cálculo forma parte del entrenamiento del modelo.

  εt = Ruido blanco, representa la parte de los datos que no se puede explicar a partir de valores pasados. Su cálculo forma parte del entrenamiento del modelo.

  p = Cantidad de valores pasados que se usan para estimar el valor presente. Se trata de un hiperparámetro del modelo. Suele fijarse entre 0 y 2.

- MA (Moving Average o Media Móvil): De manera análoga a la componente autorregresiva, el modelo considera los errores pasados para corregir el valor actual, se construye mediante una combinación lineal de los mismos:

$$y_t = c + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \cdots + \theta_q\varepsilon_{t-q} \quad (5)$$

y_t = valor de la serie en un momento determinado.

c = Constante. Su cálculo forma parte del entrenamiento del modelo.

εt = Ruido blanco, representa la parte de los datos que no se puede explicar a partir de valores pasados. Su cálculo forma parte del entrenamiento del modelo.

θ_n = Coeficiente de error pasado n, representa el grado de influencia de cada error pasado. Su cálculo forma parte del entrenamiento del modelo.

ε_{t-n} = Errores cometidos en el pasado. (diferencia entre valor estimado en el pasado y valor real).

q = Cantidad de valores pasados que se usan para estimar el valor presente. Se trata de un hiperparámetro del modelo. Suele fijarse entre 0 y 2.

- I (Integrado): Tanto la componente AR como Ma requieren que la serie sea estacionaria para funcionar correctamente, es decir, que su media sea constante, su varianza sea constante y no exista tendencia. Para poder cumplir esta condición, es necesario aplicar una transformación llamada "diferenciación" que consigue convertir la serie en estacionaria y que consiste en restar a cada valor, su valor anterior.
  De esa manera se puede observar una diferenciación de primer orden (d=1):

$$y'_t = y_t - y_{t-1} \quad (6)$$

de segundo orden (d=2):

$$y''_t = y'_t - y'_{t-1} \quad (7)$$

Y así sucesivamente, siendo d un hiperparámetro del modelo que suele fijarse entre 0 y 2.

Si combinamos estas tres componentes, obtenemos el modelo ARIMA completo, con sus tres hiperparámetros (p, d, q):

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (8)$$

Sin embargo, el modelo ARIMA solo es válido para series temporales que no tengan estacionalidad, mientras que la serie objetivo sí que posee estacionalidad, como se puede observar en la figura 18. Para poder trabajar con este escenario, es necesario añadir la componente de estacionalidad al modelo ARIMA, resultado en el modelo SARIMA (Seasonal ARIMA).

La lógica de dicha estacionalidad es similar a la del modelo ARIMA, pero los intervalos de tiempo entre series para la parte estacional vienen definidos por el periodo estacional m, que representa la duración de un ciclo completo. Dado que la serie temporal objetivo es semanal y un año contiene 52 semanas, el valor del hiperparámetro m es 52.

El resto de las componentes estacionales del modelo se calculan de forma análoga al modelo ARIMA, pero con saltos de tiempo de m. Por ejemplo, para realizar una diferenciación de primer orden, en lugar de tomarse el valor anterior en la serie, se toma el valor alejado m saltos en la serie, o, en este caso, el valor de esa misma semana el año anterior.

De esta forma, los hiperparámetros para la componente estacional del modelo SARIMA quedan definidos por letras mayúsculas como (P, D, Q, m).

En resumen, el modelo SARIMA queda definido por los siguientes hiperparámetros:

$$\text{ARIMA} \quad \underbrace{(p,d,q)}_{\text{Parte no estacional del modelo}} \quad \underbrace{(P,D,Q)_m}_{\text{Parte estacional del modelo}} \quad (9)$$

2.3.3. Prophet

Prophet es un modelo desarrollado por Facebook para pronosticar series temporales con componentes de tendencia, estacionalidad y eventos excepcionales. Su diseño modular lo hace especialmente adecuado para datos empresariales donde las interrupciones externas como promociones o vacaciones afectan significativamente las tendencias (Taylor & Letham, 2017). Prophet permite un ajuste automático de sus parámetros, lo que lo hace accesible incluso para usuarios no especializados, facilitando su adopción en diversos sectores comerciales (Carpenter et al., 2020).

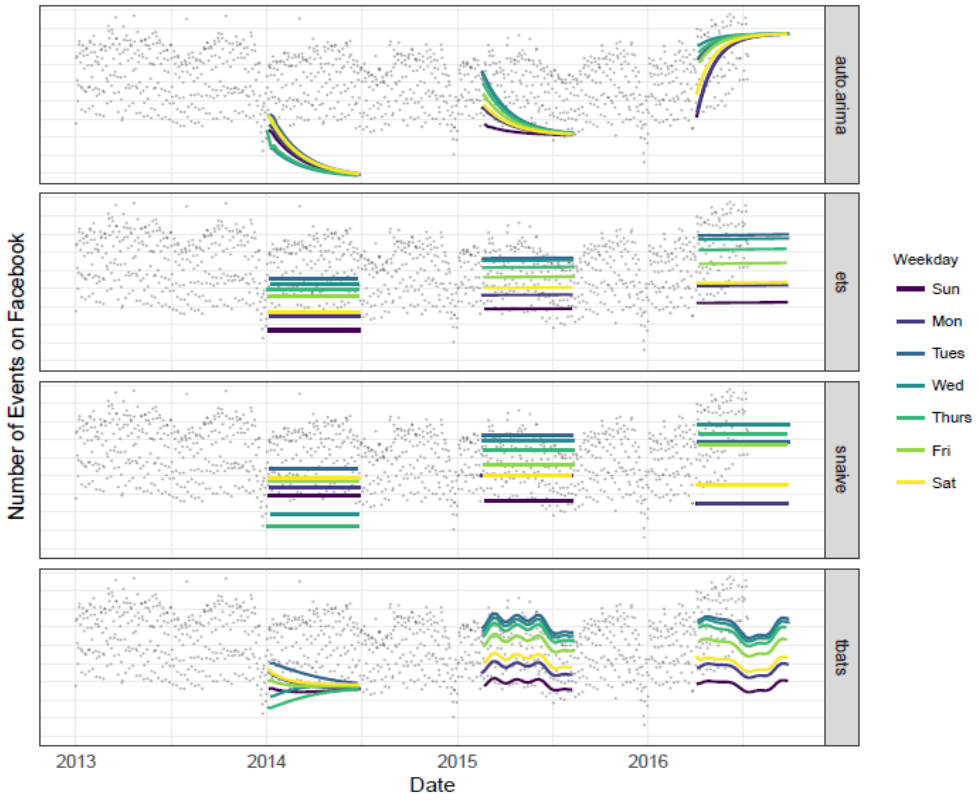
El algoritmo Prophet fue desarrollado por Sean J. Taylor y Benjamin Letham en el seno de la compañía Facebook (actual Meta) en el año 2017, este algoritmo surge como respuesta a las limitaciones de los modelos clásicos como Exponential Smoothing o ARIMA. Algunas de estas limitaciones son:

1. Dificultad para adaptarse a series temporales complejas y no estacionarias, lo que implica transformaciones adicionales.
2. Dificultad para la interpretación y el ajuste para analistas no especializados.
3. Largo tiempo de ajuste iterativo de hiperparámetros, especialmente en el modelo SARIMA.

Esta dificultad puede apreciarse en la figura 5, donde se observan las distintas predicciones de algoritmos clásicos (Como auto.arima o ETS-Exponential Smoothing) para varias series temporales.

Figura 5:

Predicción errónea del número de eventos en Facebook con distintos algoritmos.



Fuente: (Taylor & Letham, 2017)

El algoritmo Prophet emplea un enfoque distinto a los modelos vistos previamente en este apartado, ya que prioriza la facilidad en su uso y el autoajuste, así como la velocidad de entrenamiento. De esta manera consigue abstraer al analista de los principios de funcionamiento del modelo, democratizando en gran medida el análisis de series temporales.

Por otro lado, en lugar de tratar de predecir los valores futuros basándose en un modelo lineal de los valores anteriores, realiza una descomposición de la serie en tres componentes, Tendencia (g), estacionalidad (s) y efectos de eventos o días festivos (h), de forma similar a como se muestra en la figura 18.

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t. \quad (10)$$

- Tendencia (g): En el caso de la serie temporal a estudiar la tendencia de interés es la lineal, que se caracteriza porque en ella existe un crecimiento o decrecimiento constante en el tiempo. Sin embargo, el modelo permite la inserción manual o

automática de puntos de cambio de tendencia donde la tasa de crecimiento o decrecimiento varía. De esta forma la tendencia queda descrita como:

$$g(t) = (k + \mathbf{a}(t)^{\intercal}\boldsymbol{\delta})t + (m + \mathbf{a}(t)^{\intercal}\boldsymbol{\gamma}) \quad (11)$$

Donde k es la tasa de crecimiento, a(t) es un vector que controla los puntos de cambio, δ es un vector que contiene ajustes en la tasa de crecimiento, m es el valor inicial de la tendencia y γ es un vector de ajustes para asegurar la continuidad de la función en cada punto de cambio.

Esta componente puede ajustarse con el hiperparámetro “changepoint_prior_scale”, que controla la sensibilidad del modelo a los cambios de tendencia.

- Estacionalidad (s): La estacionalidad en el algoritmo Prophet está diseñada para ser capaz de capturar múltiples patrones cíclicos dentro de la serie, como pueden ser los semanales, mensuales y anuales. Para ello, el modelo utiliza series de Fourier, que permiten descomponer la serie en diversos patrones complejos. La estacionalidad queda representada como:

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right) \quad (12)$$

Donde P es el periodo de la estacionalidad, N es el número de términos de Fourier usados en la serie y a y b son dos hiperparámetros que ponderan los términos seno y coseno de la serie y que se ajustan durante el entrenamiento.

Esta componente puede ajustarse con el hiperparámetro “seasonality_prior_scale” que controla la importancia de los componentes estacionales en el modelo.

- Eventos (h): Los eventos son fechas donde se dan unas condiciones especiales que tienen un gran impacto en la serie y que se conocen de antemano, la forma más frecuente de evento es el día festivo.

El impacto de los eventos en Prophet se modela según la siguiente ecuación:

$$h(t) = Z(t)\boldsymbol{\kappa}. \quad (13)$$

Donde Z(t) es una matriz de indicadores binarios que contiene columnas correspondientes a cada evento, mientras que κ es el vector de impacto de cada evento (se asume que siguen una distribución normal).

Esta componente puede ajustarse con la variable “holidays”, que contiene las fechas específicas donde hay eventos.

2.3.4. XGBoost

XGBoost (Extreme Gradient Boosting) es un algoritmo de machine learning basado en árboles de decisión que optimiza la precisión a través de técnicas de boosting. A pesar de no ser estrictamente un algoritmo de predicción de series temporales, es conocido por su capacidad para manejar datos estructurados y no lineales, siendo una herramienta versátil para la predicción de series temporales cuando se incluyen características derivadas como retrasos y tendencias (Chen & He, 2019). En el contexto industrial, XGBoost ha sido utilizado para optimizar la planificación de recursos y mejorar la precisión en predicciones complejas (Ke et al., 2017). Su buen desempeño en una gran cantidad de problemas de regresión ha hecho que sea una de las librerías más utilizadas actualmente en el ámbito del machine learning.

XGBoost fue introducido en 2016 por Tianqi Chen y Carlos Guestrin (XGBoost Developers, 2024) y supuso una revolución en el ámbito del Machine Learning, se trata de una implementación optimizada de la técnica de boosting de gradientes que emplea además la regularización para prevenir el sobreajuste, siendo por tanto una herramienta versátil que puede ser empleada para la predicción de series temporales.

El término “boosting de gradiente” hace referencia a un método donde múltiples modelos débiles (generalmente arboles de decisión) se entrenan en secuencia, de manera que cada modelo mejora el desempeño del anterior. Para ello, se define la función objetivo, que mide como de bien el modelo ha sido capaz de ajustar los datos:

$$\text{obj}(\theta) = L(\theta) + \Omega(\theta) \quad (14)$$

Donde L(θ) es el término de pérdida o error de entrenamiento, que mide lo bien que el modelo se ajusta a los datos de entrenamiento. Normalmente se emplea como L el error cuadrático medio (MSE):

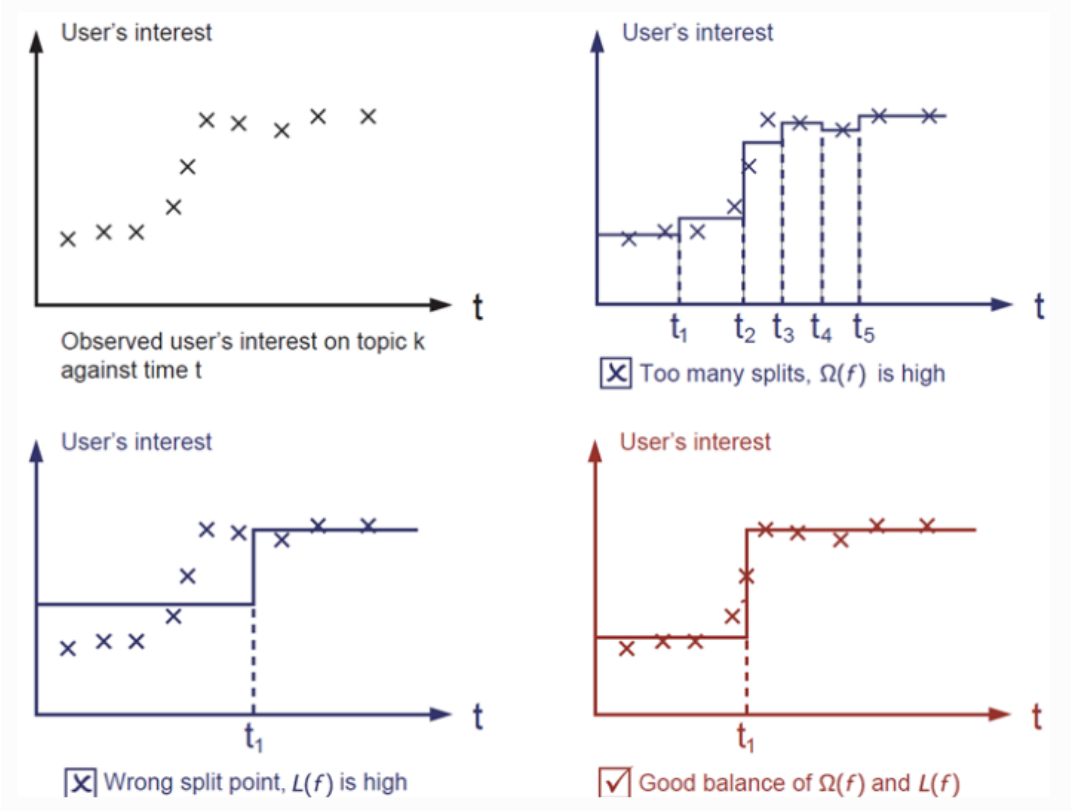
$$L(\theta) = \sum_i (y_i - \hat{y}_i)^2 \quad (15)$$

Por otra parte, Ω(θ) es el término de regularización, que controla la complejidad del modelo penalizando estructuras demasiado complejas que puedan dar lugar a sobreajuste.

En la figura 6 se explica la relación entre L y Ω para ajustar una función escalón. Como puede observarse, un modelo demasiado complejo da lugar a un Ω alto, mientras que el modelo que mejor ajusta los datos tiene un buen balance entre L y Ω.

Figura 6:

Efecto de distintos L y Ω al ajustar un modelo de escalón.



Fuente: XGBoost Developers, 2024

Sin embargo, XGBoost emplea múltiples árboles para realizar la regresión, por tanto, la predicción final se calcula como el sumatorio de cada uno de los árboles por independiente.

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F} \quad (16)$$

Mientras que la función objetivo, es de forma análoga, la suma de los dos términos de las funciones objetivo de los árboles que componen el modelo.

$$\text{obj}(\theta) = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \omega(f_k) \quad (17)$$

En este sentido se puede establecer un claro paralelismo entre los Random Forest y el modelo XGBoost, su principal diferencia radica en cómo se entrenan (Kumar, 2023).

En el modelo XGBoost se usa una estrategia aditiva: cada árbol corrige al árbol anterior y se van añadiendo de uno en uno, el resultado es una combinación de múltiples árboles donde cada uno refina las predicciones de los anteriores. Para decidir cómo dividir un nodo en el árbol, XGBoost examina todas las posibles divisiones y elige la que mejora más el rendimiento. Además, usa técnicas de "poda" (pruning), eliminando divisiones que no mejoran el modelo.

Por otro lado, la complejidad del modelo queda definida como:

$$\omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \quad (18)$$

Donde T representa el número de hojas del árbol, w_j representa el peso de dicha hoja en la predicción mientras que λ y γ son dos parámetros de regularización.

De esta forma, los principales hiperparámetros del modelo XGBoost que se van a optimizar se muestran en la tabla 2:

Tabla 2:

Hiperparámetros para XGBoost.

Hiperparámetro	Descripción	Posibles valores
objective	Objetivo de aprendizaje que define la función de pérdida para el modelo. Se fija como error cuadrático medio.	reg:squarederror
eval_metric	Métrica de evaluación para medir el rendimiento del modelo. Se fija como raiz del error cuadrático medio.	rmse
learning_rate	Tasa de aprendizaje que controla el impacto de cada árbol en la actualización del modelo.	[0.01, 0.05, 0.1]
max_depth	Profundidad máxima de los árboles de decisión.	[5, 10, 15]
subsample	Proporción de datos de entrenamiento utilizados para construir cada árbol.	[0.7, 0.8, 0.9]
colsample_bytree	Proporción de características seleccionadas al azar para construir cada árbol.	[0.7, 0.8, 0.9]
n_estimators	Número de árboles o iteraciones de boosting en el modelo.	[100, 500, 1000]
reg_alpha	Regularización L1, penaliza la magnitud de los pesos para controlar el sobreajuste.	[0, 0.1, 1]
reg_lambda	Regularización L2, penaliza la magnitud de los pesos para controlar el sobreajuste.	[0.1, 1, 10]
early_stopping_rounds	Número de iteraciones sin mejora antes de detener el entrenamiento del modelo.	50

2.3.5. Redes neuronales LSTM

Las redes neuronales LSTM (Long Short-Term Memory) son una variante de las redes neuronales recurrentes (RNN) diseñadas para aprender dependencias a largo plazo en series temporales. Introducidas por Hochreiter y Schmidhuber, las LSTM son particularmente efectivas para capturar patrones en datos secuenciales con ruido o dependencias complejas (Hochreiter & Schmidhuber, 1997). Su capacidad para procesar secuencias largas las hace ideales para aplicaciones en predicciones de series temporales no lineales y con múltiples características (Goodfellow, Bengio, & Courville, 2016).

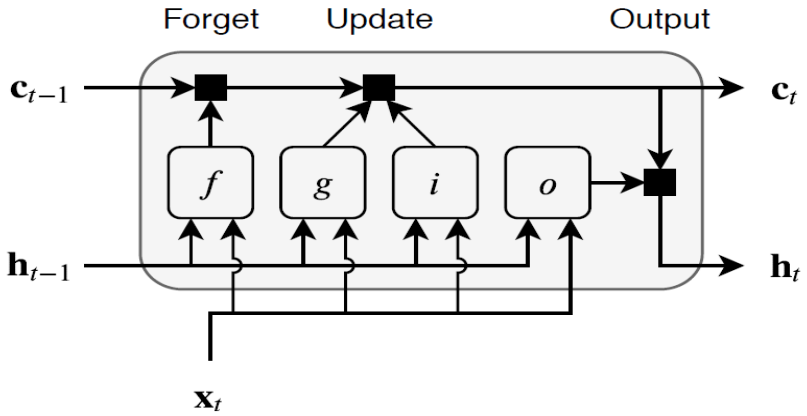
A lo largo de esta sección se explicarán brevemente los fundamentos de las redes neuronales LSTM, sin embargo, se dará por supuesto que el lector conoce como funciona una red neuronal sencilla ya que no es el propósito de este trabajo el dar una explicación detallada de las redes LSTM, sino el realizar una implementación funcional dentro del contexto empresarial.

Una red neuronal LSTM (Long Short Term Memory) es una red neuronal que es capaz de "recordar" un dato relevante en la secuencia y preservarlo por varios instantes de tiempo, es decir, posee una memoria a corto plazo (como las Redes Neuronales Recurrentes) y también una memoria a largo plazo. Estas redes son capaces de añadir o eliminar información que consideren relevante dentro de una secuencia de valores.

Para poder almacenar datos a corto y largo plazo, las redes LSTM poseen una arquitectura especial de celdas que incluyen mecanismos de "puertas" que permiten controlar el fujo de información, esta arquitectura puede verse en la figura 7.

Figura 7:

Estructura de una celda LSTM. Fuente: Mathworks



Las redes LSTM están compuestas por celdas como la mostrada en la figura 7, y cada una de estas celdas posee las siguientes puertas.

Puerta de olvido (f -Forget): Su función es decidir qué parte de la información almacenada en el estado anterior (C_{t-1}) debe eliminarse dependiendo de su relevancia y su utilidad en los siguientes pasos.

Puerta de entrada (i): Esta puerta controla que información nueva procedente de la entrada actual (X_t) y del estado oculto anterior ((h_{t-1}) se añadirá al estado de la celda. Aquí, la capa genera información candidata (g) para actualizar el estado y la puerta de entrada regula cuanta de esta información debe integrarse en la celda.

Puerta de salida (o): Esta puerta decide que parte del estado de la celda actualizado (C_t) se usará para generar la salida actual (h_t) y también que parte se pasará al siguiente paso en la secuencia.

2.4.Conclusiones

Como conclusión cabe decir que, pese a que el pronóstico de series temporales es una técnica ampliamente desarrollada, todavía no se utiliza de manera general en ambientes industriales del sector del mueble. Es de esperar que esta situación cambie en los próximos años con el auge de la IA y el machine learning en el entorno de la industria 4.0 (Telefónica y Siemens, 2025)

La orientación de este trabajo es la de demostrar la viabilidad de estas tecnologías mediante la creación de un producto mínimo viable que permita a la empresa ahorrar costes relacionados con el stock, la mano de obra y otras ineficiencias.

3. Objetivos concretos y metodología de trabajo

A lo largo de este apartado se describen los objetivos generales y específicos del proyecto, así como la metodología de trabajo que se ha empleado para la realización de este.

3.1. Objetivo general

Desarrollar un sistema predictivo avanzado mediante series temporales que permita estimar a medio plazo la carga de trabajo en cada línea de la fábrica, para poder planificar de manera más eficiente los recursos humanos y materiales necesarios para la producción.

3.2. Objetivos específicos

Los objetivos específicos desglosan los diferentes apartados que debe contener el proyecto para llegar a la consecución del objetivo general. Dichos objetivos específicos son los siguientes:

1. Procesar los datos del ERP (SAP) para garantizar una representación precisa de la carga semanal en cada línea de fabricación, permitiendo su posterior análisis.

2. Modelar las tendencias en la carga de trabajo mediante distintos modelos de series temporales para identificar patrones y seleccionar el modelo predictivo más adecuado.

3. Automatizar las predicciones de carga de trabajo y desarrollar una interfaz gráfica interactiva para permitir que los usuarios exploren los resultados y mejoren la toma de decisiones en tiempo real.

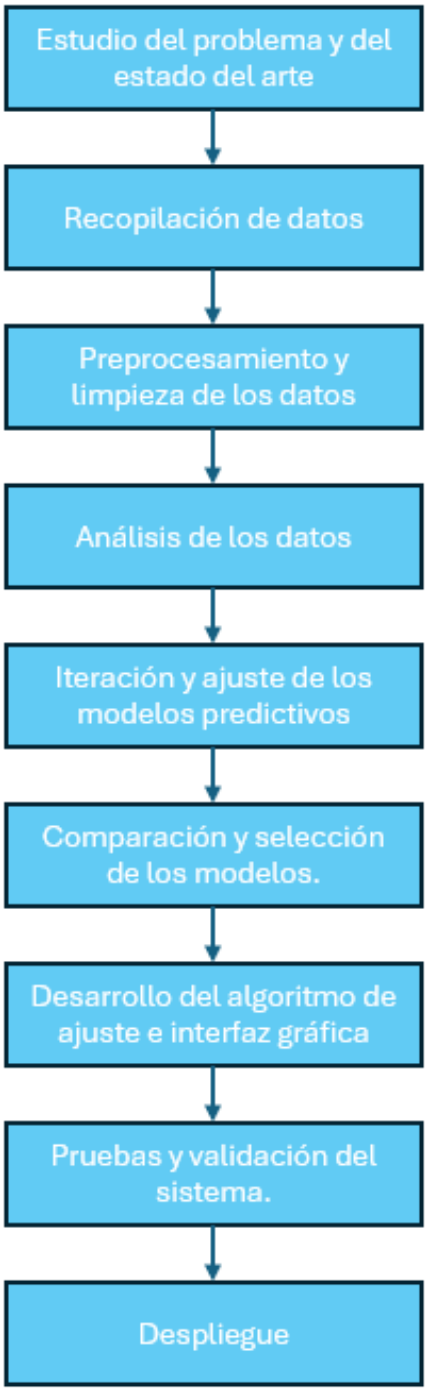
3.3. Metodología del trabajo

Para el desarrollo de este proyecto se ha adoptado una metodología ágil basada en Scrum, dada la necesidad de iterar y realizar modificaciones tanto en el código como en el alcance del proyecto en función de los resultados obtenidos. Esta metodología permite el ajuste rápido ante cualquier cambio en los requisitos del proyecto a medida que se avanza en el desarrollo del mismo.

Las fases del proyecto quedan descritas en la figura 8:

Figura 8:

Fases del proyecto



Fuente: elaboración propia

1. Estudio del problema y del estado del arte.

A lo largo de esta fase se estudia el conjunto de soluciones y modelos existentes actualmente para resolver problemas similares a los abordados en este proyecto. Se revisan trabajos de investigación, bibliografía y otras referencias que son de utilidad a la hora de realizar el desarrollo de la aplicación.

2. Recolección de datos.

Los datos que se van a emplear para su posterior análisis provienen de los escaneos que se realizan en cada línea de fabricación de la fábrica. Estos son almacenados en el ERP de la empresa (SAP) y pueden ser accedidos manualmente mediante la herramienta de SAP o automáticamente a través de una query. La recolección inicial de datos para el entrenamiento de los modelos se va a realizar mediante la herramienta de SAP de la empresa.

3. Preprocesamiento y limpieza de los datos.

Los datos en bruto han de ser tratados para su limpieza. También es necesario realizar diversos cálculos y agrupaciones, que permitan que los datos representen correctamente la carga semanal de trabajo para cada línea de fabricación. De esta forma se consigue un dataset listo para ser analizado y que puede ser empleado para entrenar modelos predictivos.

4. Análisis de los datos.

Una vez los datos han sido limpiados y pretratados, es necesario realizar un EDA (análisis exploratorio de datos) que arroje las primeras conclusiones acerca de la información contenida en los mismos. A lo largo de esta fase se estudiará el proceso productivo desde el punto de vista de los datos de fabricación.

5. Iteración y ajuste de los modelos predictivos.

Tras el análisis de los datos, se realiza un ajuste de diferentes modelos predictivos con el objetivo de estimar su fiabilidad a la hora de predecir los valores futuros de la serie temporal. Para llevar a cabo dicho ajuste fino, se modifican los hiperparámetros de cada modelo hasta encontrar los valores óptimos que maximizan la precisión de cada modelo.

6. Comparación de los modelos y selección de los modelos predictivos más convenientes.

   Estando los modelos ya entrenados y evaluados se realiza una comparación entre los mismos para estudiar su comportamiento empleando medidas de precisión de la predicción como el RMSE (Error cuadrático medio). Una vez comparados, se elegirán los modelos más convenientes para emplear en producción.

7. Desarrollo del algoritmo de ajuste automático y la interfaz gráfica.

   Elegidos ya los modelos que se van a emplear para realizar la predicción general, se desarrolla un algoritmo que sea entrenado automáticamente con los datos de producción de cada línea de fabricación y de la planta en su conjunto. Adicionalmente, se desarrolla una interfaz gráfica que permita al usuario final interactuar con el modelo predictivo, la tecnología empleada para esta labor es Streamlit.

8. Pruebas y validación del sistema.

   Una vez el software está operativo, se realizarán las pruebas pertinentes para comprobar que los resultados entregados por el modelo son coherentes y correctos. En caso de que exista alguna incoherencia en los resultados finales, se retornará al punto 7 para revisar el algoritmo y realizar los ajustes necesarios.

9. Despliegue y presentación del sistema.

   Finalmente se realizará el despliegue del software en producción y se presentará a los usuarios finales dentro de la empresa.

Respecto a la metodología de investigación, en este trabajo se ha optado por una metodología cuantitativa, ya que se basa en la recopilación, análisis y modelado de datos numéricos provenientes de la producción en una industria manufacturera. Es por esto por lo que se emplean métodos estadísticos y matemáticos para analizar series temporales, se usan métricas cuantitativas como el RMSE y el MAE para evaluar los modelos predictivos, y se realiza una comparación numérica de los diferentes algoritmos para seleccionar el más adecuado.

4. Marco normativo

En el desarrollo de este Trabajo de Fin de Máster no se han utilizado datos personales identificados ni identificables. El análisis se ha basado exclusivamente en datos de producción proporcionados por una empresa, como productos y cantidades, que no permiten ni directa ni indirectamente la identificación de personas físicas.

A pesar de no tratar datos personales, se han considerado los principios generales de confidencialidad y seguridad de los datos empresariales, siguiendo buenas prácticas en la gestión de información sensible. Las acciones llevadas a cabo incluyen:

1. Definición del alcance del tratamiento de datos:

Los datos tratados se limitan a información operativa y productiva, como:

- Tipos de productos fabricados.
- Cantidades producidas.
- Fechas y horarios relacionados con los procesos productivos.

2. Finalidad del uso de los datos:

Los datos han sido utilizados exclusivamente para calcular la carga productiva en planta y predecirla en base a los valores históricos de la serie temporal.

3. Medidas de seguridad:

Aunque no se trata de datos personales, se han aplicado medidas de protección para garantizar la integridad y confidencialidad de los datos, tales como:

- Restricción de acceso a los datos durante el desarrollo del proyecto.
- Almacenamiento en entornos seguros.
- Eliminación de los datos tras finalizar el análisis, conforme a los requisitos del proyecto.

4. Documentación del proceso:

Se ha mantenido un registro claro y transparente sobre el origen, el propósito y el uso de los datos, de acuerdo con los principios de trazabilidad y responsabilidad proactiva.

Estas prácticas están alineadas con las recomendaciones generales para la gestión ética de datos empresariales, como se destaca en los principios de buenas prácticas de la OCDE (2013) para la seguridad de la información y la privacidad de los datos no personales. Asimismo, se ha considerado la importancia de seguir medidas de protección alineadas con estándares como la ISO 27001, que garantiza un enfoque sistemático para la seguridad de la información (ISO, 2022).

Dado que no se han empleado datos personales, no se ha requerido la aplicación del Reglamento General de Protección de Datos (RGPD) ni de la Ley Orgánica 3/2018 (NLOPD). Sin embargo, se han seguido prácticas éticas en el manejo de los datos empresariales para garantizar su correcto uso durante el desarrollo del TFM.

5. Desarrollo específico de la contribución

A lo largo de este capítulo se explica la solución desarrollada para dar respuesta al problema descrito en los capítulos anteriores, así como los resultados obtenidos.

5.1. Requisitos técnicos

En esta sección se definen los requisitos con los que ha de contar la solución desarrollada en el proyecto, los elementos con los que debe contar el sistema y las limitaciones del mismo que quedan fuera del alcance de este proyecto. Se trata de una fase crítica ya que a lo largo de la misma quedan preestablecidas las características que el usuario final o cliente espera del sistema y cuál va a ser el grado de cumplimiento de las mismas por parte del desarrollador.

De este modo, los requisitos técnicos que se han definido para este proyecto son los siguientes:

- El servicio se ejecutará en un servidor local de la empresa y será usado por los departamentos de ingeniería, materiales y producción.

- El sistema ha de ser capaz de establecer una medida de la cantidad de trabajo en cada línea de la fábrica y en la planta completa (idealmente en horas de trabajo).

- Para el desarrollo del sistema será necesario realizar una selección de los modelos más convenientes para este tipo de problema.

- Es necesario que la solución ingiera los datos del ERP de la empresa y realice las predicciones automáticamente, incluyendo el ajuste de los hiperparámetros para cada modelo.

- Las medidas de efectividad de la predicción (RMSE y MAE) han de ser visibles para cada modelo.

- Debe existir una interfaz gráfica que permita interactuar con la herramienta seleccionando la serie temporal (línea de fabricación) que se desea visualizar y el modelo que se desea usar para realizar la predicción.

- Ha de poderse visualizar la predicción de todos los modelos simultáneamente para poder realizar comparaciones entre ellos.

- La gráfica de líneas donde se muestra la serie temporal y sus predicciones ha de ser interactiva, permitiendo hacer zoom en los valores y usar un modo de pantalla completa para explorar mejor los valores.

- Ha de poder verse el valor numérico de la serie al pasar el ratón por encima de esta.

5.2. Tecnologías empleadas

A lo largo de esta sección se describen brevemente las distintas tecnologías que se han empleado para el desarrollo de este proyecto.

1. **Python:**

   El lenguaje de programación Python ha sido el núcleo del desarrollo del proyecto debido a su flexibilidad y al amplio ecosistema de bibliotecas especializadas del que dispone. Este lenguaje de programación ha sido utilizado tanto para la preparación y manipulación de los datos como para la implementación de modelos predictivos de series temporales. Su facilidad de uso y su comunidad activa han facilitado la integración de distintas herramientas y el acceso a recursos actualizados. Además, su enfoque en la legibilidad permitió un desarrollo ágil y organizado a lo largo del proyecto. (Python Software Foundation, 2023).

2. **Pandas:**

   La librería Pandas ha sido esencial para la manipulación y el análisis de datos en este proyecto. Su capacidad para gestionar DataFrames ha permitido estructurar los datos de forma eficiente, realizando su limpieza y transformación. Además, su funcionalidad de indexación temporal y agrupamiento facilitó el análisis de tendencias y estacionalidades en las series temporales. La integración con otras bibliotecas, como

NumPy y Matplotlib, ha asegurado un flujo de trabajo ágil en todas las etapas del desarrollo (McKinney, 2022).

3. **Seaborn y Matplotlib:**

   La visualización de datos desempeña un papel crucial en la fase de análisis exploratorio y en la comunicación de los resultados obtenidos por cada uno de los modelos predictivos. Para poder visualizar los datos que se están tratando se han empleado dos librerías de gran popularidad en el entorno Python: Seaborn y Matplotlib.

   Seaborn: Esta biblioteca permitió generar gráficos estadísticos de alta calidad con un diseño atractivo y coherente. Ha sido especialmente útil para representar correlaciones, distribuciones y patrones en las series temporales (Waskom, 2022).

   Matplotlib: Complementariamente, Matplotlib ha ofrecido un control total sobre los detalles de los gráficos generados con Seaborn, posibilitando la creación de visualizaciones personalizadas para destacar aspectos específicos de los datos o de los resultados de los modelos. Esta combinación de herramientas ha ayudado a identificar patrones, tendencias y estacionalidades en los datos iniciales y a interpretar las predicciones generadas (*Matplotlib Developers*, 2023).

4. **Streamlit:**

   Streamlit ha sido la solución seleccionada para desarrollar una interfaz web interactiva y sencilla que permite al usuario final explorar los resultados de los modelos de manera intuitiva y accesible. Esta herramienta destaca por su simplicidad en la implementación de aplicaciones web a partir de scripts de Python, lo que ha permitido integrar visualizaciones dinámicas y controles personalizables para cargar las predicciones de distintos modelos (Streamlit Inc., 2023).

5.3. Tratamiento de datos y EDA

A lo largo de esta sección se va a describir en detalle el estudio y la transformación de los datos desde el punto de partida inicial, hasta la obtención de un dataset listo para realizar el entrenamiento de los diferentes modelos que se evalúan en la sección 5.4.

5.3.1. Dataset de partida.

Como se ha explicado a lo largo del capítulo 3, el objetivo de este proyecto es el de ofrecer una solución capaz de predecir la demanda en las distintas líneas de la fábrica, para poder adelantarse a la misma y de este modo mejorar la eficiencia de la planta.

Cada vez que un producto es fabricado en la planta, la orden de trabajo de dicho producto se escanea con un scanner industrial. Dicho escaneo queda registrado en el ERP de la empresa junto con distintos datos, como la fecha y hora de escaneo, planta donde se ha fabricado el producto, código de scanner que ha realizado el registro, pedido al que pertenece el registro, tipo de artículo etc.

En la figura 9 puede observarse uno de los scanner industriales utilizados para la adquisición de datos de fabricación.

Figura 9:

Proceso de escaneo de la orden de trabajo del producto final.

Para poder estimar la carga de trabajo de la planta, con la ayuda del departamento de IT se ha extraído del software ERP de la empresa (SAP) el dataset en bruto mostrado en la tabla 3:

Tabla 3:

Dataset de escaneos de productos en planta (Dataset de partida)

Nodos	ZEX_RF_WH_HISTOR-SDATE	ZEX_RF_WH_HISTOR-STIME	Creado por	CódT	Unidad manipulación	Entrega	Pos.	Doc.ventas	Pos.	Material	Denominación
MADR	01.12.2023	7:03:20	SAPRFMADR16	ZRF_M7	2316183522	90651697	190	1305311	41700	1404848001	ARM US3 PTA PERS
MADR	01.12.2023	7:05:31	SAPRFMADR20	ZRF_M7	2316183469	90644929	150	1323299	18300	P1072550	DIVISIO FRAMELESS SCREEN
MADR	01.12.2023	7:06:31	SAPRFMADR20	ZRF_M7	2316183523	90644929	150	1323299	18300	P1072550	DIVISIO FRAMELESS SCREEN
MADR	01.12.2023	7:06:51	SAPRFMADR16	ZRF_M7	2316183524	90651697	60	1305311	17600	1404848001	ARM US3 PTA PERS
MADR	01.12.2023	7:06:53	SAPRFMADR20	ZRF_M7	2316183525	90644929	150	1323299	18300	P1072550	DIVISIO FRAMELESS SCREEN
MADR	01.12.2023	7:07:34	SAPRFMADR19	ZRF_M7	2316183526	90646134	10	1329944	300	P1072551	DIVISIO FRAMELESS SCR - DESK CONNECTORS
MADR	01.12.2023	7:07:50	SAPRFMADR20	ZRF_M7	2316183527	90644929	150	1323299	18300	P1072550	DIVISIO FRAMELESS SCREEN
MADR	01.12.2023	7:08:25	SAPRFMADR19	ZRF_M7	2316183528	90652505	20	1334687	1100	P1072551	DIVISIO FRAMELESS SCR - DESK CONNECTORS
MADR	01.12.2023	7:08:38	SAPRFMADR20	ZRF_M7	2316183511	90644929	150	1323299	18300	P1072550	DIVISIO FRAMELESS SCREEN
MADR	01.12.2023	7:09:05	SAPRFMADR16	ZRF_M7	2316183592	90651697	60	1305311	17600	1404848001	ARM US3 PTA PERS
MADR	01.12.2023	7:09:11	SAPRFMADR19	ZRF_M7	2316183593	90652505	20	1334687	1100	P1072551	DIVISIO FRAMELESS SCR - DESK CONNECTORS
MADR	01.12.2023	7:09:56	SAPRFMADR19	ZRF_M7	2316183594	90652505	20	1334687	1100	P1072551	DIVISIO FRAMELESS SCR - DESK CONNECTORS
MADR	01.12.2023	7:09:58	SAPRFMADR20	ZRF_M7	2316183595	90644929	10	1323299	16200	P1072550	DIVISIO FRAMELESS SCREEN
MADR	01.12.2023	7:10:13	SAPRFMADR20	ZRF_M7	2316183470	90644929	10	1323299	16200	P1072550	DIVISIO FRAMELESS SCREEN
MADR	01.12.2023	7:10:31	SAPRFMADR16	ZRF_M7	2316183471	90651697	60	1305311	17600	1404848001	ARM US3 PTA PERS
MADR	01.12.2023	7:10:31	SAPRFMADR20	ZRF_M7	2316183612	90644929	10	1323299	16200	P1072550	DIVISIO FRAMELESS SCREEN
MADR	01.12.2023	7:10:48	SAPRFMADR20	ZRF_M7	2316183613	90644929	30	1323299	16500	P1072550	DIVISIO FRAMELESS SCREEN
MADR	01.12.2023	7:11:14	SAPRFMADR19	ZRF_M7	2316183614	90652505	20	1334687	1100	P1072551	DIVISIO FRAMELESS SCR - DESK CONNECTORS
MADR	01.12.2023	7:11:48	SAPRFMADR20	ZRF_M7	2316183615	90644929	30	1323299	16500	P1072550	DIVISIO FRAMELESS SCREEN
MADR	01.12.2023	7:11:51	SAPRFMADR26	ZRF_M1	2316183616	90650467	40	1334499	800	P616_LEG_C_FIX	PIE TIPO C ALT FIJA FUSION SOLO
MADR	01.12.2023	7:12:27	SAPRFMADR16	ZRF_M7	2316183617	90651697	60	1305311	17600	1404848001	ARM US3 PTA PERS
MADR	01.12.2023	7:12:35	SAPRFMADR19	ZRF_M7	2316183618	90652505	20	1334687	1100	P1072551	DIVISIO FRAMELESS SCR - DESK CONNECTORS

Este dataset contiene los registros de todos los escaneos que se han realizado en la fábrica a lo largo del tiempo desde el 4 de enero del año 2021 hasta el día 18 de Julio del año 2024.

Las columnas relevantes de este dataset, así como su interpretación son las siguientes:

1. **Nodos**:

   Esta columna representa la planta de fabricación donde se ha realizado el escaneo. Dado que al extraer la información del ERP se ha realizado un filtrado previo, esta columna tiene el valor constante de MADR.

2. **ZEX_RF_WH_HISTOR-SDATE**:

   Fecha de realización del escaneo, formato “dd.mm.yyyy”

3. **ZEX_RF_WH_HISTOR-STIME**:

   Hora de realización del escaneo, formato “hh:mm:ss”

4. **Creado por**:

   Scanner que ha realizado el registro. Cada scanner está relacionado con una línea distinta de fabricación. Por ejemplo, “SAPRFMADR16” representa a la línea de volúmenes (Armarios, Taquillas, Archivos…).

   Los códigos de escáneres de cada línea de fabricación se muestran en la tabla 4.

Tabla 4:

Equivalencia entre línea de fabricación y código de escáner.

Scanner	Linea de fabricación
'SAPRFMADR26'	'Mesas',
'SAPRFMADR20'	'Paneles',
'SAPRFMADR23'	'Tableros',
'SAPRFMADR16'	'Volumenes',
'SAPRFMADR19'	'Conectores',
'SAPRFMADR21'	'Varios',
'SAPRFMADR17'	'Silla DO',
'SAPRFMADR18'	'Pie Ajustable',
'SAPRFMADR15'	'Bloques',
'SAPRFMADR24'	'Carpintería otros',
'SAPRFMADR25'	'Partito Rail'

5. **Material**:

   Número de artículo que se ha escaneado, está relacionado con el producto fabricado.

6. **Denominación**:

   Descripción del artículo que se ha escaneado, se trata de una descripción del producto fabricado.

El resto de las columnas (CódT, Unidad manipulación, Entrega, Pos., Doc.ventas, Pos), representan información relacionada con el pedido y no son relevantes para este proyecto, ya que no están relacionadas con la carga de trabajo de la fábrica.

5.3.2. Limpieza de datos

Todo el trabajo descrito en las secciones posteriores se ha realizado empleando la librería Pandas sobre Python corriendo en un Jupyter Notebook.

En primer lugar, se ha importado el dataset. Este se encuentra en tres archivos diferentes: “01.21-12.22.xlsx”, “12.22-12.23.xlsx” y “12.23-07.24.xlsx”. Esto se debe a que los datos de mayor antigüedad se almacenan en una base de datos de archivo, mientras que los datos más recientes se almacenan en una base de datos denominada histórica.

Tras concatenar los tres archivos para formar el dataset de partida, se procede a eliminar 2837 registros duplicados que han sido encontrados en el mismo. Tras ello, se han renombrado las columnas para que tengan nombres más intuitivos, como “fecha_arch”, “hora_arch”, y se han eliminado los espacios antes y después de los String.

Para las columnas que representan horas, como ”hora_arch” (perteneciente a la base de datos de archivo) y “hora_hist” (perteneciente a la base de datos histórica), se han transformado sus datos a variables de tipo “timedelta”, para después combinarlas en una sola columna “hora”.

Se ha calculado el mes y el año del escaneo a partir de la fecha completa y se han añadido en dos columnas adicionales

Por último, se ha calculado el turno de fabricación en el que se ha realizado el escaneo, sabiendo que el turno de mañana (turno 1) es el que contiene las piezas escaneadas antes de las 15:15, mientras que el turno de tarde es el que contiene las piezas escaneadas después. Para ello se ha definido una función específica “calcular_turno”. Esta información será crucial a la hora de estimar la carga de trabajo de cada producto.

En la tabla 5 puede observarse una muestra del dataset de partida una vez limpiado y pretratado:

Tabla 5:

Datos de partida tras limpieza y pre-tratamiento.

	nodo	scanner	codigo	unidad_	entrega	posicion	doc_vent	posicion	material	denominacion	fecha	hora	turno	mes	year
0	MADR	SAPRFMADR16	ZRF_M7	2314925272	89172516	20	9712112	30	P845_USTORAGE_HGD	USNG ARMARIO P/BAT	44200	0 days 07:06:35	1	1	2021
1	MADR	SAPRFMADR19	ZRF_M7	2314925292	89174813	50	9711027	70	P1072551	DIVISIO FRAMELESS SCR - DI	44200	0 days 07:09:18	1	1	2021
2	MADR	SAPRFMADR19	ZRF_M2	2314925273	89170341	10	9708990	30	P1072551	DIVISIO FRAMELESS SCR - DI	44200	0 days 07:11:45	1	1	2021
3	MADR	SAPRFMADR23	ZRF_M1	2314925312	89172410	20	9710338	70	P5SP_PCUS86468	TOP A1600 P800 W/TA ACTI	44200	0 days 07:12:32	1	1	2021
4	MADR	SAPRFMADR23	ZRF_M1	2314925313	89172410	20	9710338	70	P5SP_PCUS86468	TOP A1600 P800 W/TA ACTI	44200	0 days 07:12:35	1	1	2021
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1481130	MADR	SAPRFMADR26	ZRF_M1	2316412904	96027592	70	1411736	2000	P5SP_1900434	Band 1/2 bench 1600	45491	0 days 22:42:58	2	7	2024
1481131	MADR	SAPRFMADR26	ZRF_M1	2316412905	96027592	70	1411736	2000	P5SP_1900434	Band 1/2 bench 1600	45491	0 days 22:43:34	2	7	2024
1481132	MADR	SAPRFMADR26	ZRF_M1	2316412906	96027592	70	1411736	2000	P5SP_1900434	Band 1/2 bench 1600	45491	0 days 22:44:30	2	7	2024
1481133	MADR	SAPRFMADR26	ZRF_M1	2316412907	96027592	70	1411736	2000	P5SP_1900434	Band 1/2 bench 1600	45491	0 days 22:47:32	2	7	2024
1481134	MADR	SAPRFMADR26	ZRF_M1	2316412908	96027592	70	1411736	2000	P5SP_1900434	Band 1/2 bench 1600	45491	0 days 22:48:34	2	7	2024
1479039 rows × 15 columns															

5.3.3. Análisis exploratorio de datos (EDA).

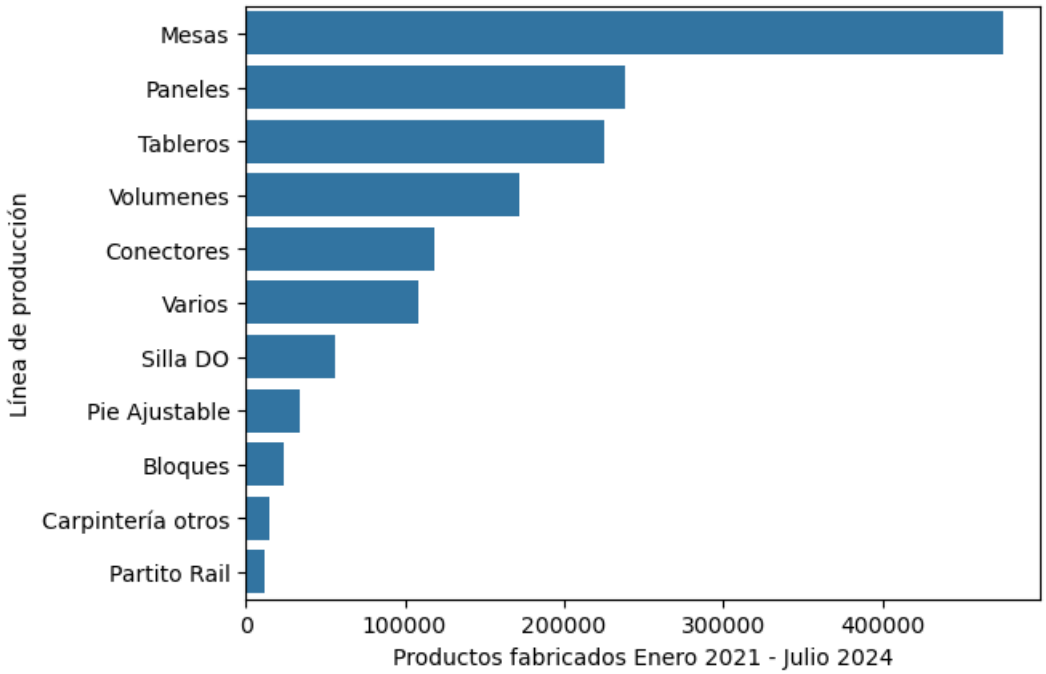
5.3.3.1. EDA general de planta.

Una vez listos los datos de todos los escaneos de planta desde enero de 2021, es posible realizar un análisis exploratorio de datos (EDA) de los datos de partida, con el objetivo de tratar de comprender en profundidad como se distribuye la fabricación de productos en la planta.

En primer lugar, se realiza un conteo simple de la cantidad de productos fabricados en la planta desde enero de 2021 hasta Julio de 2024 divididos por líneas de fabricación, la gráfica de barras resultante se puede observar en la figura 10.

Figura 10:

Productos fabricados en cada línea de fabricación

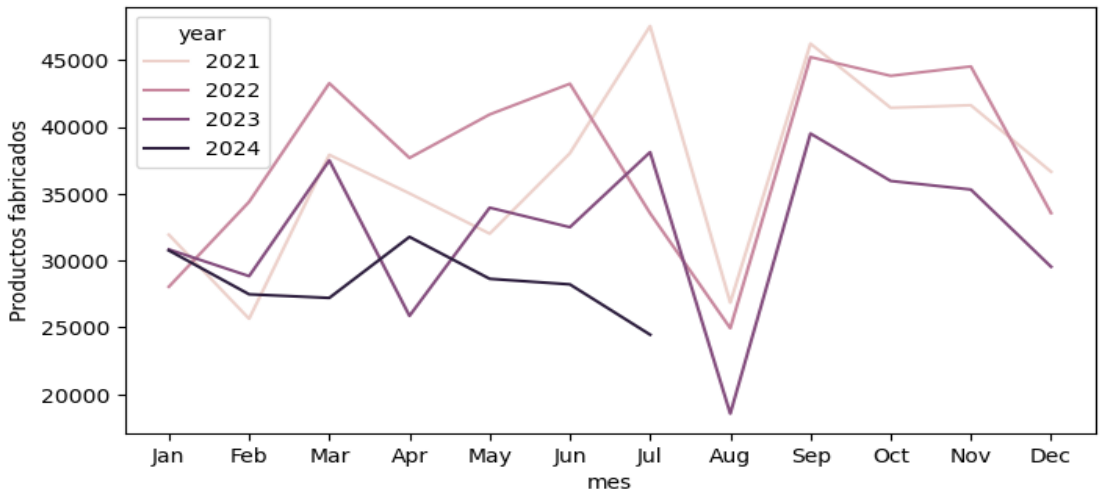


Como puede observarse, la línea de fabricación de mesas (patas de mesa) es la que mayor cantidad de productos fabrica respecto al resto de líneas. Es importante remarcar que eso no necesariamente significa que esa línea sea la que mayor cantidad de horas de trabajo emplea, ya que la cantidad de minutos de trabajo requeridos para fabricar cada producto puede ser menor en esa línea en comparación con otras.

En la figura 11 puede observarse un gráfico de líneas con la evolución de la cantidad de productos fabricados a lo largo del mes en cada uno de los años desde enero de 2021 hasta Julio de 2024.

Figura 11:

Evolución de productos fabricados. Años 2021-2024



Es fácil observar que a lo largo del año se dan patrones estacionales que se repiten. Por ejemplo, el volumen de productos fabricados se reduce en agosto mientras que se incrementa en Septiembre (alcanzando frecuentemente el máximo anual). También se observa un decremento en diciembre-enero, que continua hasta febrero, fecha en la que vuelve a incrementar la producción. Se observa una tendencia general decreciente, acrecentada a partir del año 2023, debido a un cambio en el modelo de negocio de la empresa.

La existencia de una clara estacionalidad y tendencia en los datos, observables ambas a simple vista a partir de un análisis exploratorio visual, es indicativo de que estos datos pueden ser predichos a partir de su serie histórica por un modelo estadístico o de machine learning.

5.3.3.2. EDA de línea de montaje de volúmenes.

Una vez estudiado el comportamiento de la demanda general de la empresa, es el momento de centrarse en una de las líneas de fabricación con el objetivo de evaluar su comportamiento por independiente.

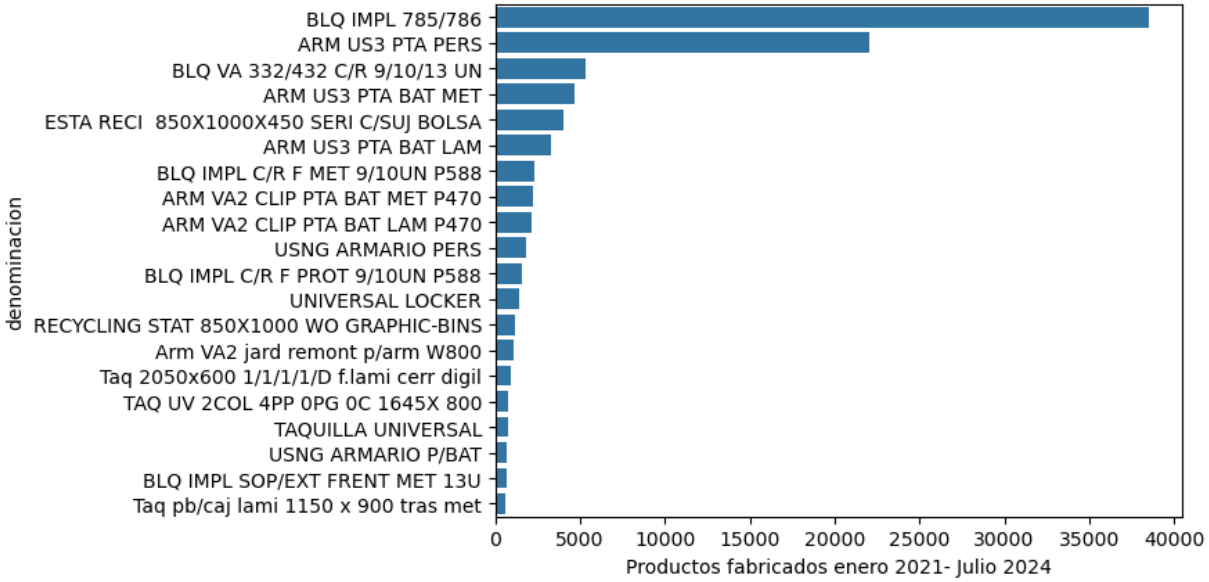
En este caso, se ha decidido centrarse en la línea de volúmenes, donde se fabrican productos como armarios, taquillas y archivos. Pese a que esta no es la línea que más volumen de trabajo tiene por cantidad de productos terminados, sí que es la línea que más trabajo da a otras

secciones de la fábrica, ya que la cantidad de componentes necesarios para elaborar un armario es muy superior a la cantidad de componentes necesarios para fabricar una mesa o un panel divisor.

En la figura 12, puede observarse un gráfico de barras con los distintos productos fabricados en la línea de volúmenes entre enero de 2021 y julio de 2024, así como su cantidad:

Figura 12:

Productos y cantidades fabricadas en la línea de volúmenes. Años 2021-2024



Como puede observarse en la figura 12, la mayor parte de los productos que se fabrican en la línea son “Bloques” (archivadores pequeños con ruedas) y Armarios. Sin embargo, en total se han fabricado 644 referencias distintas en esta línea. Esto concuerda con el principio de Pareto, que aplicado en este caso expresa que la mayor parte de la capacidad se emplea en una pequeña variedad de productos, mientras que la gran mayoría de productos representan la minoría de la capacidad productiva empleada.

5.3.4. Medición de la cantidad de trabajo.

Esta es una de las principales contribuciones del presente trabajo al estado del arte, ya que es preciso realizar una medición adecuada de la carga de trabajo en planta que posteriormente pueda ser estimada. Esto no se trata de una cuestión trivial, dada la gran cantidad de productos diferentes que se fabrican en esta planta, cada uno de ellos con sus particularidades y carga de trabajo.

Una primera opción podría ser medir la cantidad de trabajo en planta según los productos fabricados (número de escaneos). Esta aproximación presenta un problema fundamental ya que los productos son muy diferentes entre sí, y cada uno de ellos requiere una cantidad de trabajo (minutos) diferente. Por ejemplo, la fabricación de una taquilla requiere una gran cantidad de minutos de trabajo en la línea mientras que la fabricación de un bloque requiere menos de un minuto de trabajo.

De esta forma, en ingeniería industrial de procesos se define el concepto de Takt Time, como el ritmo de fabricación de un producto en la línea de producción:

$$Takt\ Time_{Producto} = Ritmo\ de\ fabricación\ _{Producto}$$

Por ejemplo, si la línea de fabricación produce una taquilla cada 3 minutos, se dice que el Takt Time de las taquillas es de 3 minutos.

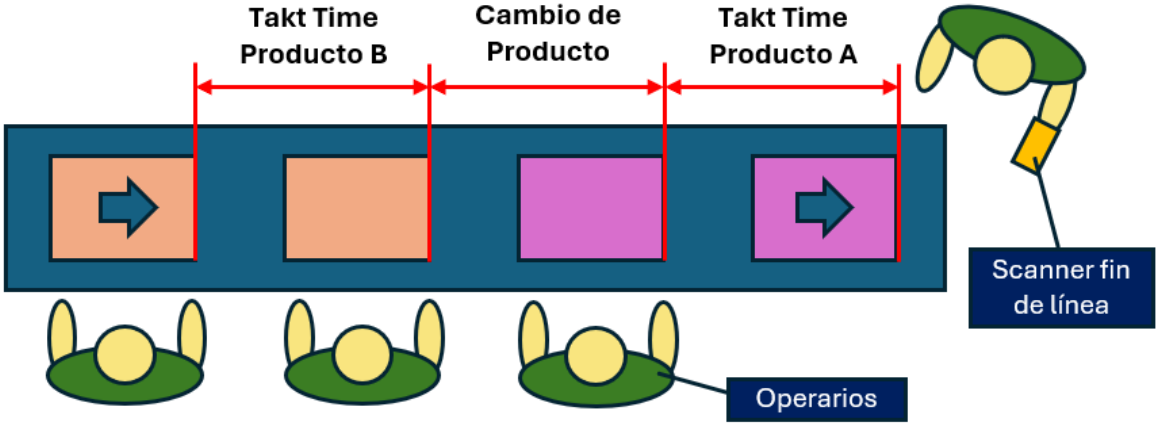
Por tanto, si se multiplica el Takt Time de un producto, por la cantidad de productos que se han fabricado, se obtiene el tiempo total de trabajo real en la línea para fabricar ese producto.

$$Tiempo\ Efectivo\ de\ trabajo\ en\ línea\ _{Producto} = Takt\ Time\ _{Producto} * Cantidad\ Fabricada\ _{Producto}$$

Es importante diferenciar el tiempo de trabajo “Efectivo” del tiempo de trabajo total de la línea, ya que este último puede incluir descansos, cambios de utillaje y otras ineficiencias.

Figura 13:

Línea de producción y Takt Time



En la figura 13 puede verse la representación de una línea de producción en cadena con distintos operarios. Como puede observarse, el Takt Time es el tiempo que transcurre entre la finalización de la producción de un producto y el siguiente. Si se produce un cambio de producto, el tiempo de ese cambio también será medido. Cada uno de esos eventos queda registrado mediante la operación de escaneo de la hoja de trabajo.

Por lo tanto, el Takt Time de cada producto puede calcularse como la diferencia en segundos entre la columna "hora" de escaneo de un producto y el siguiente. Sin embargo, es importante realizar tres filtrados antes de calcular el Takt Time medio para cada producto:

1. Filtrado de productos con Takt Time superior a 600 segundos:

   Este Takt Time no representa el tiempo de fabricación de un producto, sino un descanso que se ha realizado en la línea (se ha escaneado un producto, se ha hecho una pausa y tras ella se ha escaneado el siguiente).

   Por convenio las pausas de producción son de 15 minutos. Si la pausa ha sido producida por una avería no existe un tiempo de pausa estándar, pero la experiencia demuestra que tienden a ser mayores a 10 minutos.
2. Filtrado de Takt Times de productos que se han fabricado tras un producto diferente:

   Cuando se mide el tiempo entre productos de dos productos que no son iguales, no se está midiendo solamente el tiempo de fabricación, sino que se está midiendo el tiempo de cambio de producto en la línea (esto puede implicar cambios de componentes o

utillajes). Es por eso por lo que es preciso filtrar estos Takt Times. Este efecto puede observarse en la figura 13.

3. Emplear para el cálculo, solo productos fabricados en el turno de mañana.

   El turno de mañana dura de 7:00 a 15:00 y se caracteriza por que la cantidad de operarios en las líneas es siempre la nominal. Si para calcular el Takt Time se empleasen también productos fabricados en el turno de tarde, se estaría introduciendo una variabilidad adicional en los datos, ya que la mano de obra del turno de tarde puede variar en función de la cantidad de trabajo de la fábrica.

Una vez realizados estos filtrados, se ha calculado el Takt Time medio a lo largo de todo el dataset para cada producto fabricado en la línea de volúmenes, el resultado puede observarse en la tabla 6.

Tabla 6:

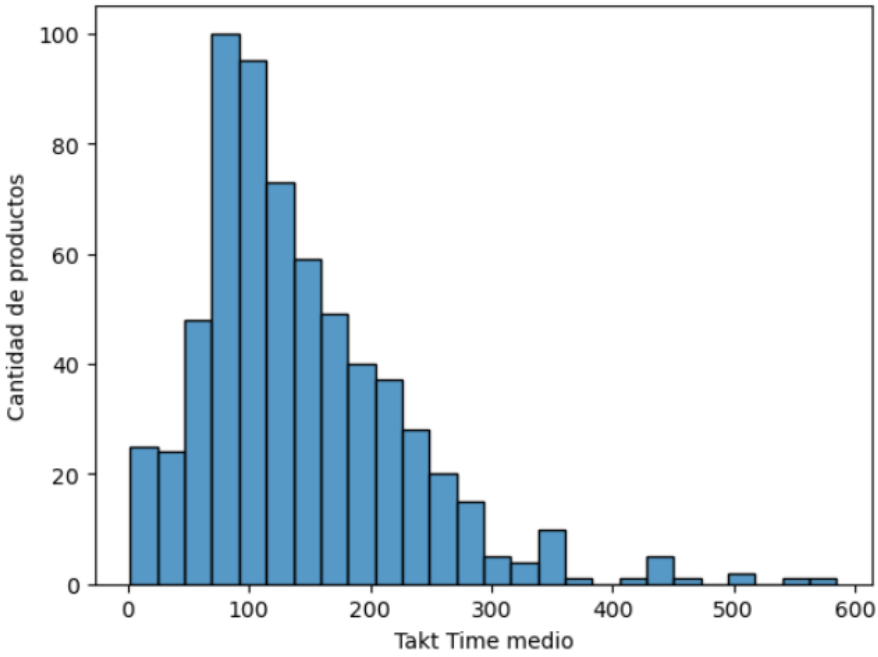
Takt Times y desviación estándar para los productos más fabricados en la línea de volúmenes

denominacion	media_takt	dev_std	fabricados
BLQ IMPL 785/786	84,3	66,4	38527
ARM US3 PTA PERS	91,2	58,2	22075
BLQ VA 332/432 C/R 9/10/13 UN	84,9	61,7	5295
ARM US3 PTA BAT MET	102,4	57,5	4665
ESTA RECI 850X1000X450 SERI C/SUJ BOLSA	86,7	49,0	3992
ARM US3 PTA BAT LAM	105,7	60,3	3264
BLQ IMPL C/R F MET 9/10UN P588	90,6	70,6	2332
ARM VA2 CLIP PTA BAT MET P470	114,6	61,7	2194
ARM VA2 CLIP PTA BAT LAM P470	114,3	63,3	2142
USNG ARMARIO PERS	88,5	63,5	1844

En la figura 14 puede observase un histograma que representa la distribución de Takt Times medios en todos los productos fabricados en la línea de volúmenes. La mayoría de los productos tienen Takt Times del orden de 100 segundos (1 minuto 40 segundos), mientras que, a medida que el Takt Time medio se incrementa, la cantidad de productos disminuye. Existen muy pocos productos que tarden más de 300 segundos (5 minutos) en ser fabricados. Por otro lado, existen unos pocos productos que se producen muy rápido (menos de 50 segundos).

Figura 14:

Histograma de Takt Times en la línea de volúmenes.



A partir del cálculo de los Takt Times (velocidad de fabricación de cada producto), es posible calcular la carga de trabajo total en un día para la línea de fabricación de la siguiente manera:

$$Carga\ de\ trabajo\ diario\ (s)_{Linea\ de\ fabricación} = \sum_{productos\ del\ dia} Takt\ Time_{Producto} * Cantidad_{Producto} \quad (19)$$

En la tabla 7 puede observarse una agrupación con la carga de trabajo en la línea de volúmenes, así como la cantidad de productos que se han fabricado a lo largo de todos los días en los cuales se dispone de datos: 04/01/2021 – 18/07/2024.

Tabla 7:

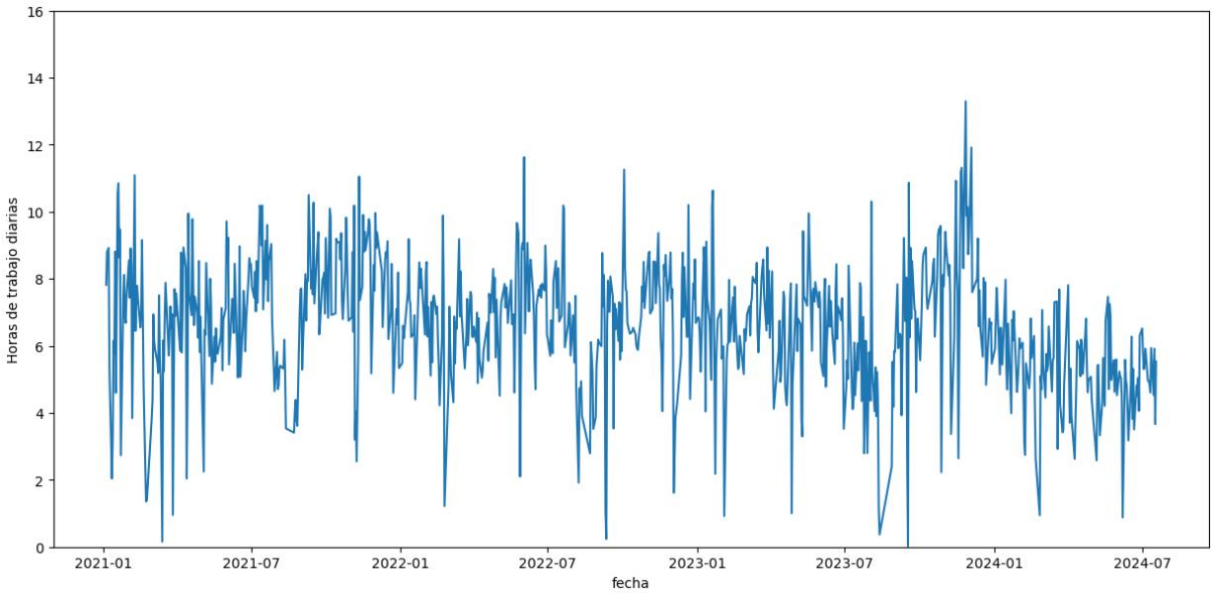
Cantidad de productos, y carga de trabajo para la línea de volúmenes

	fecha	cantidad_fabricada	carga_trabajo	carga_trabajo_horas	semana
0	04/01/2021	258	28.153	7,8	2021-01
1	05/01/2021	362	31.686	8,8	2021-01
2	07/01/2021	307	32.122	8,9	2021-01
3	08/01/2021	201	18.957	5,3	2021-01
4	11/01/2021	75	7.336	2,0	2021-02
...	...	...	...	...	...
843	12/07/2024	193	21.371	5,9	2024-27
844	15/07/2024	172	16.337	4,5	2024-28
845	16/07/2024	205	21.294	5,9	2024-28
846	17/07/2024	107	13.193	3,7	2024-28
847	18/07/2024	170	19.914	5,5	2024-28

Si realizamos una representación gráfica de la carga de trabajo en horas de la línea de volúmenes para cada día de trabajo, se obtiene el resultado mostrado en la figura 15.

Figura 15:

Serie temporal de carga de trabajo para cada día en línea de volúmenes.

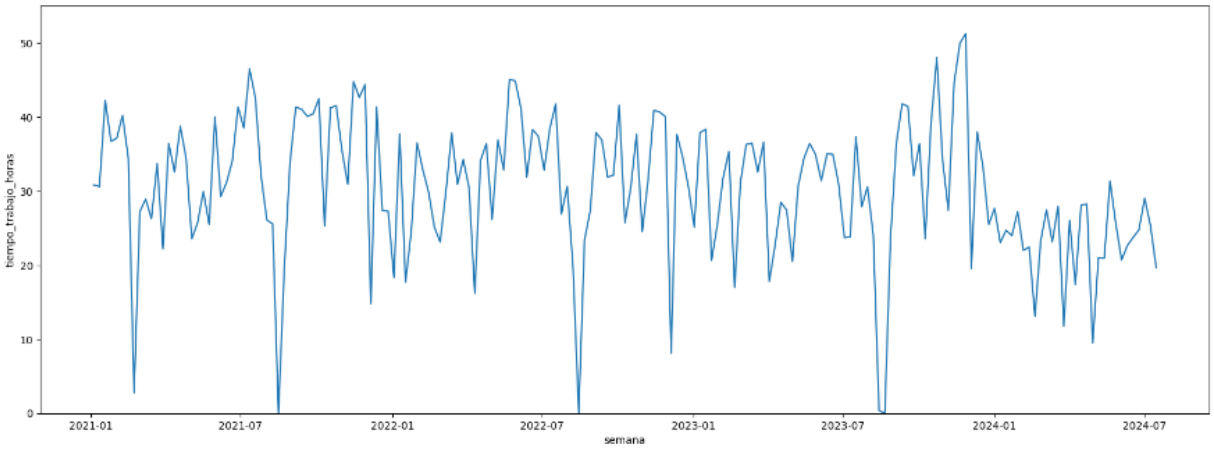


Como puede observarse en la figura 15, la serie temporal que contiene la carga de la línea para cada día presenta gran cantidad de altibajos y ruido en general.

Dado que las decisiones operativas en planta generalmente no se toman de un día para otro, se realiza una agrupación de la carga de trabajo semanal para esta misma serie temporal, obteniendo los resultados de la figura 16.

Figura 16:

Serie temporal de carga de trabajo para cada semana en línea de volúmenes.

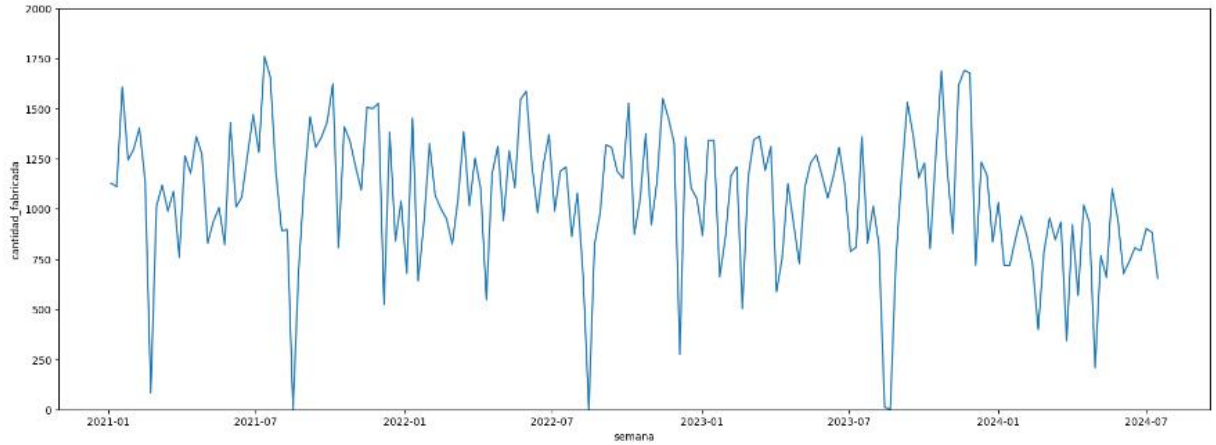


Con esta configuración la serie temporal se vuelve mucho más limpia, permitiendo identificar estacionalidades y tendencias generales de manera más sencilla. Por tanto, será esta la serie temporal que se intente predecir en la sección 5.4 y que se utilizará para comparar el rendimiento de los modelos en la sección 5.5.

Resulta interesante, además, comparar la gráfica de la figura 16, con una gráfica donde se representen simplemente la cantidad de productos fabricados cada semana, dicha representación puede observarse en la figura 17.

Figura 17:

Serie temporal de cantidad de productos para cada semana en línea de volúmenes.



Comparando las figuras 16 y 17 se identifica que ambas presentan patrones similares, sin embargo, es preciso aclarar que la serie temporal de la figura 16 representa el trabajo en planta de manera mucho más fiable, ya que puede darse el caso de que un pico en la cantidad

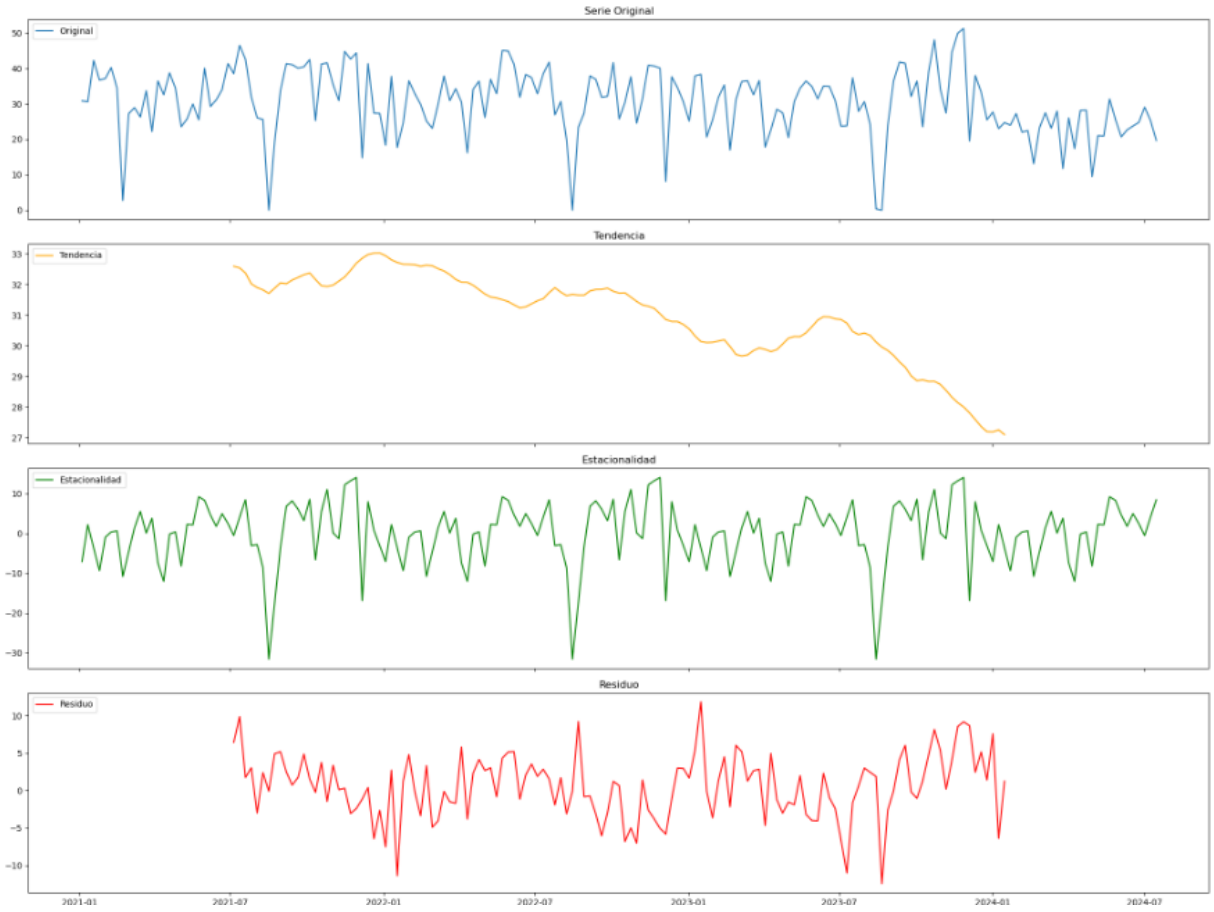
de productos a fabricar no represente un pico en la cantidad de trabajo (si dichos productos se fabrican con pocas horas de trabajo) y viceversa.

5.3.5. Estacionalidad, y tendencia de la serie temporal

En la figura 18, puede observarse la descomposición de la serie temporal objetivo (figura 16) en tres componentes principales: Tendencia, Estacionalidad y residuo (Chatfield, 2003).

Figura 18:

Tendencia, Estacionalidad y Residuo de la serie temporal de carga de trabajo semanal.



La tendencia es el cambio a largo plazo en la serie temporal, representa la dirección en la que la serie se mueve de manera general, sin tener en cuenta fluctuaciones a corto plazo. Como puede observarse, la tendencia de esta serie temporal es a la disminución en el tiempo.

Por otra parte, la estacionalidad representa los patrones regulares y repetitivos que se dan en intervalos específicos, en este caso anuales. Esta componente captura, por ejemplo, el descenso de la carga en agosto debido a la parada de mantenimiento o el pico de carga en septiembre y a finales de año.

Por último, el residuo representa la parte de la serie temporal que no se ha podido capturar a través de la tendencia y estacionalidad. Es decir, representa la parte aleatoria e impredecible de la serie. Es fundamental que todos los modelos predictivos desarrollados en el apartado 5.4 sean capaces de capturar estas tres componentes adecuadamente.

5.4. Entrenamiento y evaluación de los modelos

A lo largo de esta sección se utilizan los datos agrupados de carga de trabajo semanal (en horas de trabajo) mostrados en la figura 16 para entrenar distintos modelos de predicción. El horizonte de predicción para todos los modelos será de 16 semanas, ya que, en base a la experiencia previa de la compañía, 16 semanas son suficientes para poder anticiparse operativamente a los cambios en el mercado.

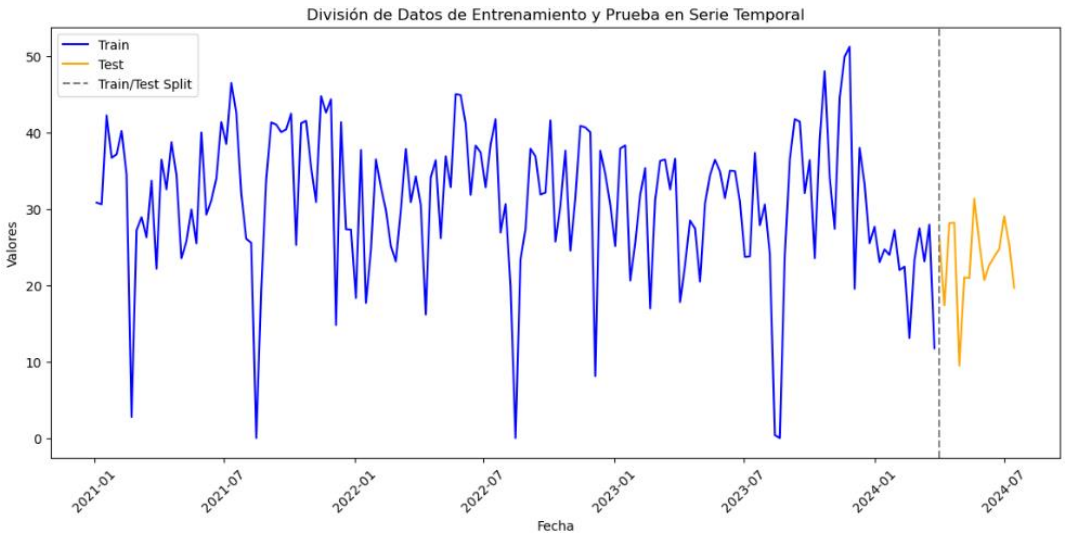
Los modelos a entrenar partirán de los más sencillos (como el Suavizado Exponencial) a los más complejos y modernos (como las redes neuronales LSTM), pudiendo comparar el funcionamiento de estos en función de su complejidad, ya que no necesariamente los modelos más complejos son los más convenientes en términos de funcionamiento.

Para cada modelo, se partirá de la explicación teórica de los fundamentos de la sección 2.3. A continuación se dividirá el dataset en un conjunto de entrenamiento y otro de test, empleando para testear el rendimiento del modelo las últimas 16 semanas de datos disponibles. Es importante remarcar que en el contexto de los problemas de series temporales los datos de entrenamiento y test no han de mezclarse de forma aleatoria, sino que deben respetar su orden temporal. Esto se debe a que la variable tiempo es la principal (y a veces única) variable de predicción para estos modelos, y es empleada para identificar tendencias y patrones estacionales en los datos.

La división de los datos entre Train y Test puede observarse en la figura 19.

Figura 19:

División de datos de entrenamiento y test.



Dado que la predicción de series temporales es un problema de regresión, el rendimiento de cada modelo se evaluará en base a las siguientes métricas:

1. RMSE: Root Mean Squared Error.

El Root Mean Squared Error se define como (Hyndman & Koehler, 2006, p. 680):

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Esta métrica se calcula como la raíz cuadrada del sumatorio de los errores (diferencia entre valor predicho y real) al cuadrado, dividido entre el número de errores. Se trata de una métrica que penaliza los grandes errores, castigando los valores atípicos, haciendo de esta la métrica más robusta (y por lo general más utilizada) para calcular los errores de una regresión.

2. MAE: Mean Absolute Error.

El Mean Absolute Error se define como (Willmott & Matsuura, 2005, p. 80):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Esta métrica se calcula como la media del valor absoluto de los errores cometidos por un modelo de regresión. Su principal ventaja es que se trata de una métrica muy intuitiva y fácil de calcular, sin embargo, no es demasiado sensible a los valores atípicos, lo que provoca que sea menos robusta que el RMSE a la hora de comprender el rendimiento de un modelo.

3. MAPE: Mean Absolute Percentage Error:

El Mean Absolute Percentage Error se define como (de Myttenaere et al., 2015, p. 39):

$$\text{Error absoluto porcentual} = \left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100$$

Esta métrica se calcula como la media del error absoluto porcentual, es decir, el porcentaje de error respecto al valor real que el modelo ha cometido. Se trata de una métrica muy fácil de interpretar pero que nuevamente, no es demasiado sensible a valores atípicos, también puede ser problemática cuando se están estimando valores cercanos a cero.

5.4.1. Exponential Smoothing – Holt Winters

Este es el primer modelo que se va a entrenar para predecir la serie temporal objetivo. Sus fundamentos han sido explicados en la sección 2.3.1.

5.4.1.1. Entrenamiento iterativo

Para tratar de estimar los hiperparámetros más convenientes para el modelo de Holt-Walters, se ha realizado una iteración para los tres hiperparámetros α, β, γ.

Para ello, se ha establecido un grid entre 0 y 1 con un salto de 0.01 para cada hiperparámetro y se ha entrenado un modelo con los datos de entrenamiento, a continuación, se han evaluado sus predicciones con los datos de test calculando el RMSE. Para cada iteración, si el valor de RMSE es menor que el mejor obtenido hasta el momento, se almacenan los hiperparámetros. De esta forma se han obtenido los siguientes resultados:

Mejor α = 0.05

Mejor β = 0.3

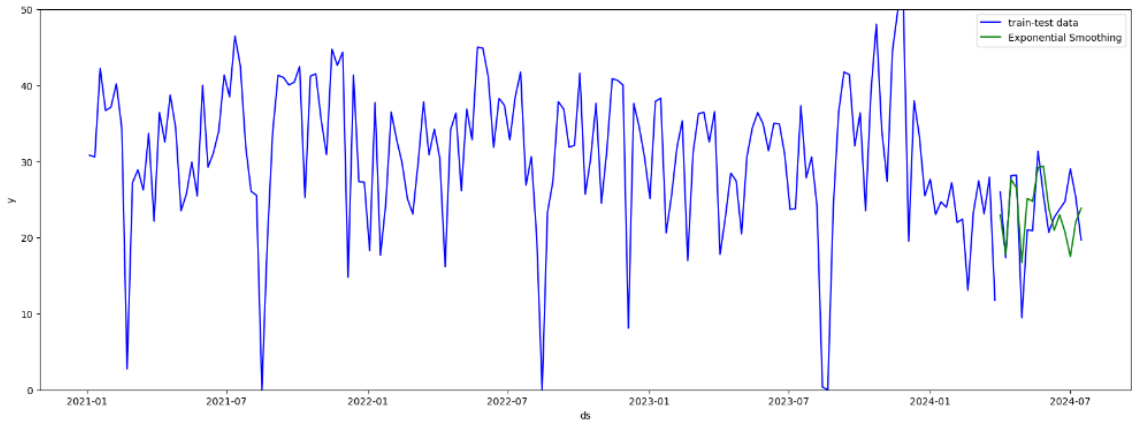
Mejor γ = 0.2

Mejor RMSE = 4.37

La predicción de los datos de test para estos hiperparámetros se muestra en la figura 20.

Figura 20:

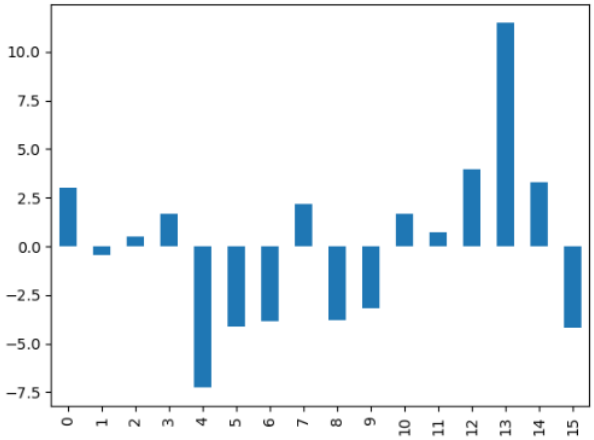
Predicción de los datos de test en Holt-Winters con ajuste iterativo de hiperparámetros



Los errores cometidos en la predicción se muestran gráficamente en la figura 21.

Figura 21:

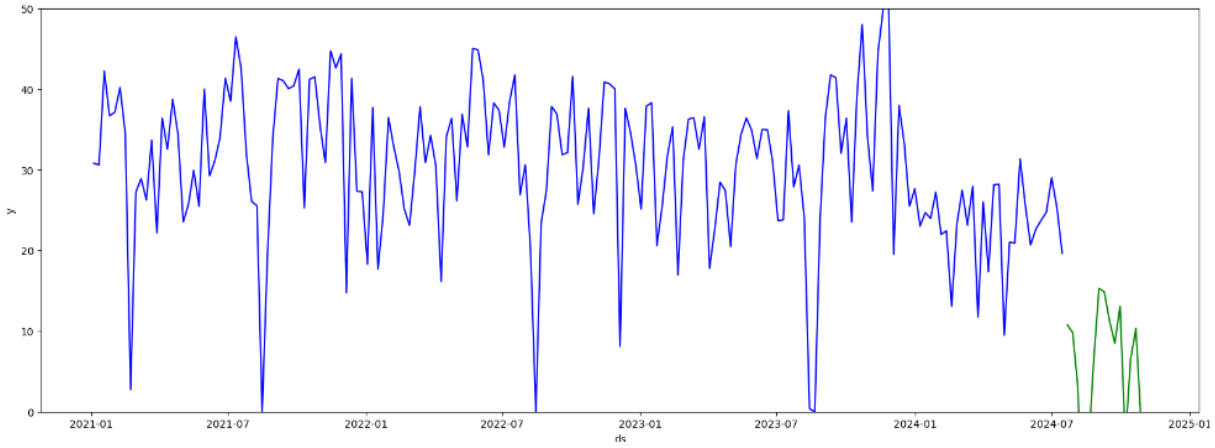
Error en la predicción de los datos de test en Holt-Winters con ajuste iterativo de hiperparámetros



Sin embargo, cuando se emplean estos hiperparámetros para la predicción de valores futuros, puede observarse que la predicción no se realiza correctamente. Esto se debe a que el ajuste iterativo de hiperparámetros ha dado lugar a un sobreajuste para los datos de entrenamiento y test, pero no ha sido capaz de capturar los patrones generales de la serie, dando lugar a predicciones carentes de sentido como se muestra en la figura 22.

Figura 22:

Predicción errónea de valores futuros usando Holt-Winters con ajuste iterativo de hiperparámetros

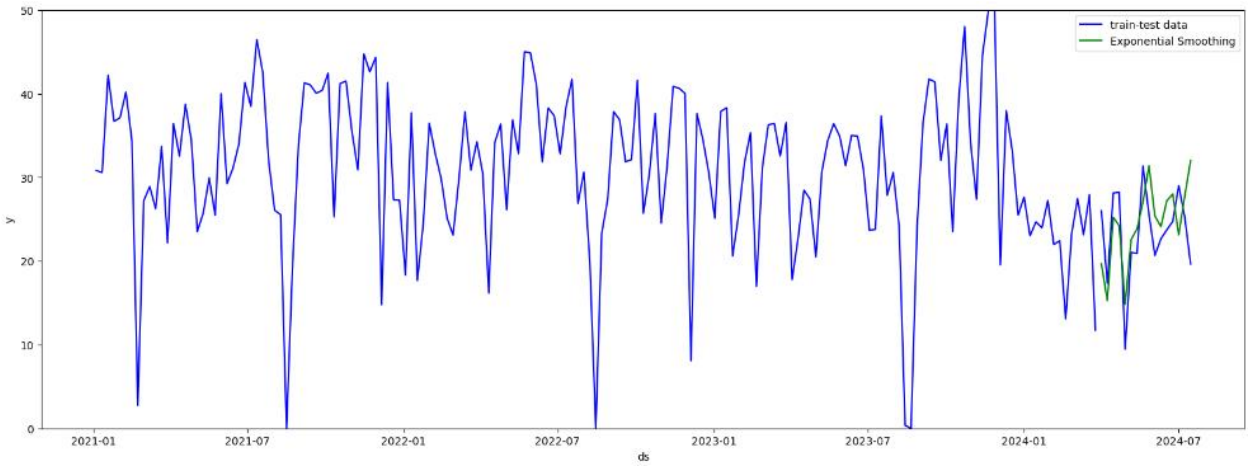


5.4.1.2. Entrenamiento automático usando el método de máxima verosimilitud (MLE).

Dado que la búsqueda iterativa de hiperparámetros ha dado lugar a sobreajuste, se decide que la librería empleada para implementar el modelo de Holt-Winters (Statsmodels) realice el ajuste de hiperparámetros de forma automática utilizando el método de máxima verosimilitud o MLE por sus siglas en inglés (Statsmodels, 2023), los resultados de dicho entrenamiento se muestran en la figura 23.

Figura 23:

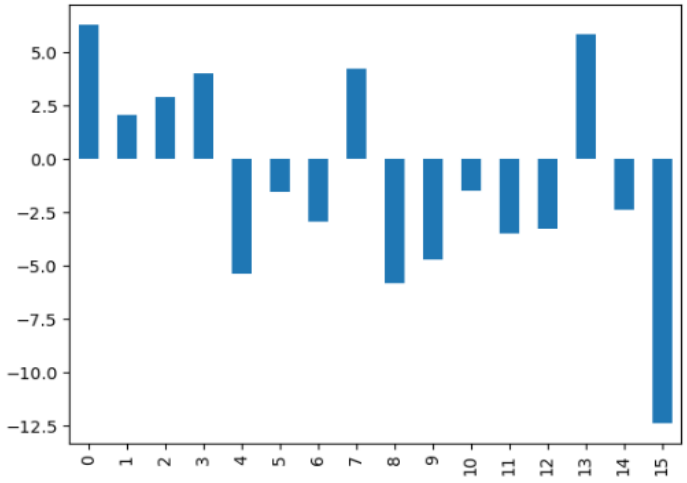
Predicción de los datos de test en Holt-Winters con ajuste automático MLE de hiperparámetros



En la figura 24 pueden observarse gráficamente los errores en la predicción de los datos de test con ajuste automático.

Figura 24:

Error en la predicción de los datos de test en Holt-Winters con ajuste automático MLE de hiperparámetros



Las métricas alcanzadas mediante este método de entrenamiento se muestran en la tabla 8.

Tabla 8:

Métricas de error para el modelo de Holt-Winters

	RMSE	MAE	MAPE
Holt-Winters	5,001	4,296	20,30%

Mientras que los valores finales alcanzados por los hiperparámetros son los siguientes:

α = 0.2013 β = 1.466 e-09 γ = 2.273 e-08

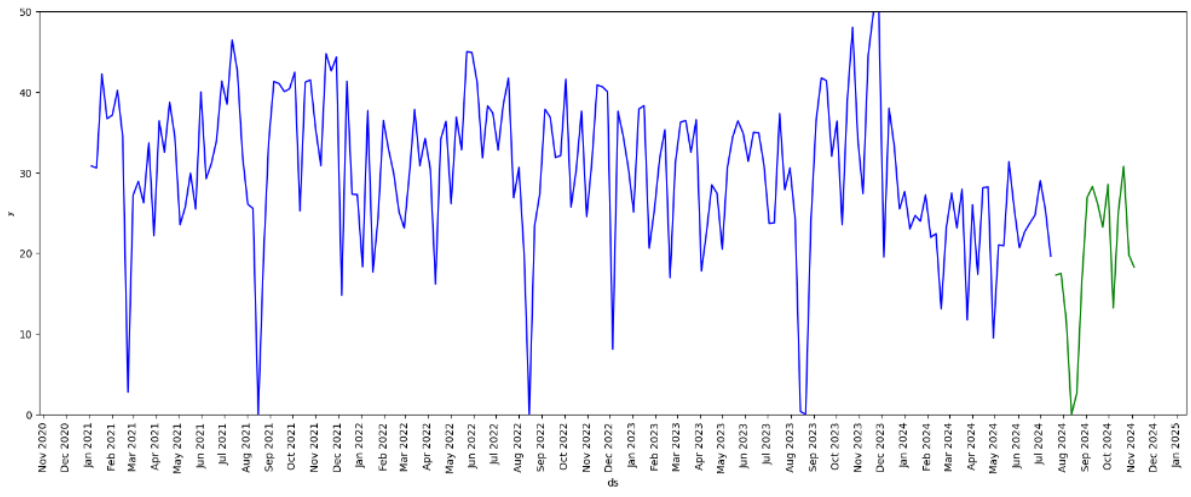
Como puede observarse en la tabla 8, el RMSE para este método de ajuste es mayor que para el ajuste iterativo. Un valor α de 0.2 significa que el modelo da relativamente más peso a los datos históricos que los recientes al calcular el nivel, es decir, que el modelo considera que la serie es estable y no necesita ajustes rápidos en función de los últimos datos observados.

Por otro lado, un valor β tan bajo sugiere que el modelo no está ajustando la tendencia activamente ya que considera que la tendencia es prácticamente inexistente o muy estable. Del mismo modo, un valor γ tan bajo sugiere que a estacionalidad es muy estable y no necesita cambios frecuentes, lo cual tiene sentido ya que la experiencia demuestra que el patrón de carga en la planta se repite anualmente de una manera similar.

La predicción final se observa en la figura 25. El modelo ha sido capaz de predecir la parada de agosto, esto indica que ha capturado adecuadamente el patrón en la serie temporal.

Figura 25:

Predicción de la carga de planta en las próximas 16 semanas con el modelo de Holt-Winters



5.4.2. SARIMA

Este es el segundo modelo que se va a entrenar para predecir la serie temporal objetivo. Sus fundamentos han sido explicados en la sección 2.3.2.

5.4.2.1. Entrenamiento iterativo

De forma análoga al entrenamiento del modelo de suavizado exponencial, en primer lugar, se ha realizado una búsqueda iterativa de los hiperparámetros (p, d, q) y (P, D, Q, m).

El grid definido ha sido de (0, 0, 0) (0, 0, 0, 52) a (3, 2, 4) (2, 0, 2, 52).

Para ello se emplea un bucle que entrena un modelo con los hiperparámetros correspondientes y los datos de train, a continuación, predice los datos de test y calcula el RMSE, almacenando los hiperparámetros que han arrojado un menor RMSE.

En la tabla 9, se pueden observar los 15 mejores valores de hiperparámetros y el RMSE que han arrojado al predecir los datos de test.

Tabla 9:

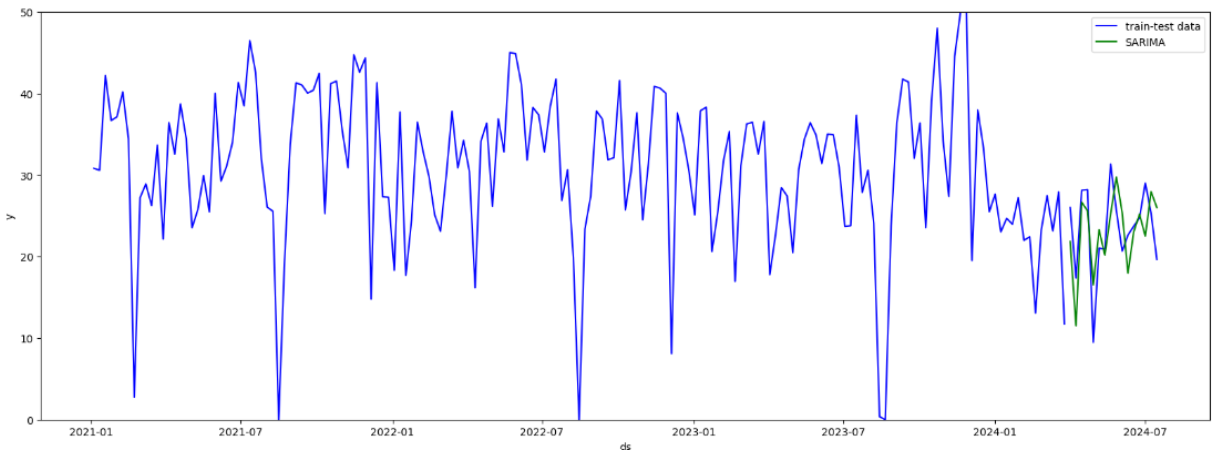
Mejores hiperparámetros para el ajuste iterativo del modelo SARIMA.

Iteración	pdq	sea_pdq	rmse
1613	(3, 2, 4)	(2, 0, 2, 52)	4,37
1208	(2, 2, 4)	(2, 0, 2, 52)	4,39
1291	(3, 0, 2)	(2, 1, 1, 52)	4,56
1292	(3, 0, 2)	(2, 1, 2, 52)	4,56
182	(0, 1, 1)	(2, 0, 2, 52)	4,57
328	(0, 2, 2)	(0, 1, 1, 52)	4,80
337	(0, 2, 2)	(1, 1, 1, 52)	4,80
1543	(3, 2, 2)	(0, 1, 1, 52)	4,80
1397	(3, 1, 1)	(2, 0, 2, 52)	4,81
209	(0, 1, 2)	(2, 0, 2, 52)	4,83
1424	(3, 1, 2)	(2, 0, 2, 52)	4,97
166	(0, 1, 1)	(0, 1, 1, 52)	4,99
617	(1, 1, 2)	(2, 1, 2, 52)	5,03
760	(1, 2, 3)	(0, 1, 1, 52)	5,04

El valor de la predicción de los datos de entrenamiento y los hiperparámetros (3, 2, 4) (2, 0, 2, 52) se muestra en la figura 26.

Figura 26:

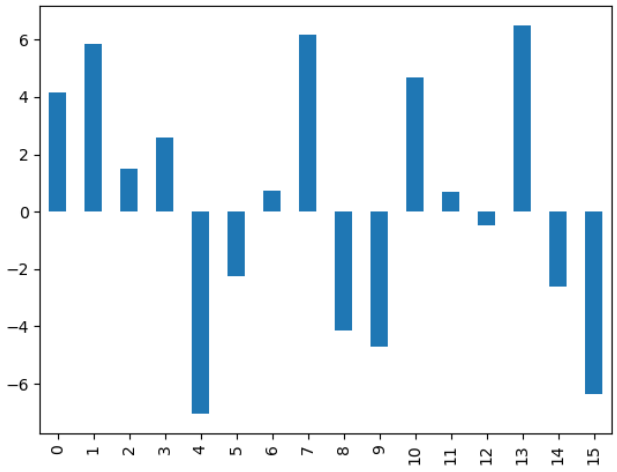
Predicción de los datos de test en SARIMA con ajuste iterativo de hiperparámetros



Los errores cometidos por el modelo se muestran gráficamente en la figura 27.

Figura 27:

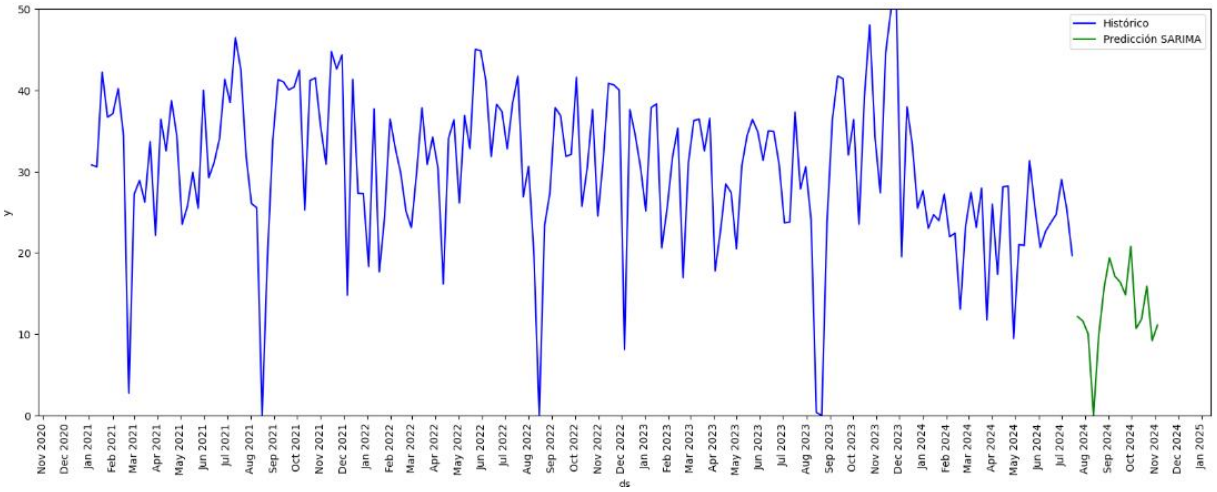
Error en la predicción de los datos de test en SARIMA con ajuste iterativo de hiperparámetros



No obstante, al estudiar las predicciones futuras alcanzadas con este método, en la figura 28 se puede observar que los valores que el modelo arroja son poco probables, ya que una caída tan drástica en la carga de trabajo no es esperable.

Figura 28:

Predicción errónea de valores futuros usando SARIMA con ajuste iterativo de hiperparámetros



Nuevamente, este efecto se debe al sobreajuste del modelo, que se especializa en los datos de entrenamiento siendo incapaz de generalizar la serie.

5.4.2.2. Entrenamiento automático.

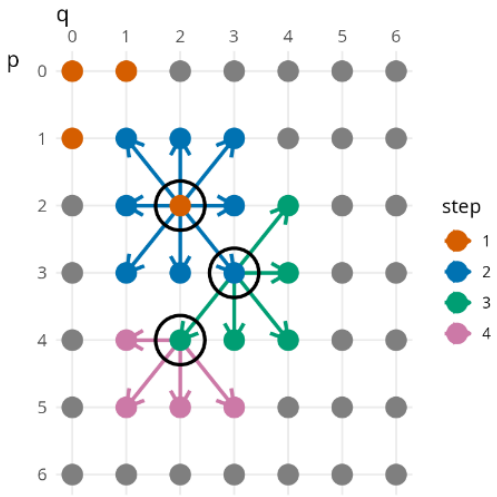
Dado que el ajuste iterativo del modelo SARIMA da lugar a sobreajuste, y el ajuste manual no es una opción, pues el objetivo final de estos algoritmos es que puedan ser implementados de forma automática en un sistema predictivo, se ha empleado la librería auto_arima para hacer una búsqueda automatizada de los mejores hiperparámetros.

La librería auto_arima emplea el algoritmo de Hyndman-Khandakar para hacer una búsqueda en un grid de todos los posibles valores de hiperparámetros que conduzca a una solución óptima. (Hyndman & Khandakar, 2008)

En la figura 29 se observa como el algoritmo busca en un espacio de hiperparámetros p-q y avanza hasta encontrar una combinación que minimice el AIC o Criterio de Información de Akaike. (Burnham & Anderson, 2002).

Figura 29:

Búsqueda en el grid de hiperparámetros p-q del mejor ajuste de SARIMA.



Fuente: Hyndman & Athanasopoulos, 2021

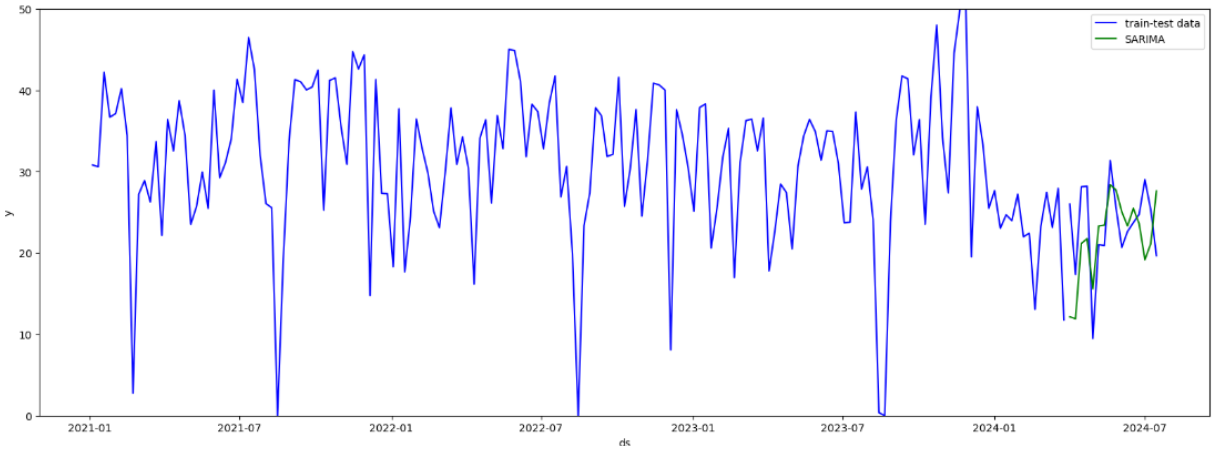
El ajuste automático de los hiperparámetros dura 362 segundos (6 minutos), lo cual puede ser un problema a la hora de implementar el algoritmo final para todas las líneas de producción, ya que es de esperar que el ajuste requiera 6 minutos por línea.

Los hiperparámetros que mejores resultados han conseguido mediante el método de autoajuste son (1, 1, 1) (2, 0, 0, 52), lo cual significa que no se ha aplicado diferenciación estacional ni existe término de promedio móvil estacional.

Las predicciones de los datos de test se muestran en la figura 30.

Figura 30:

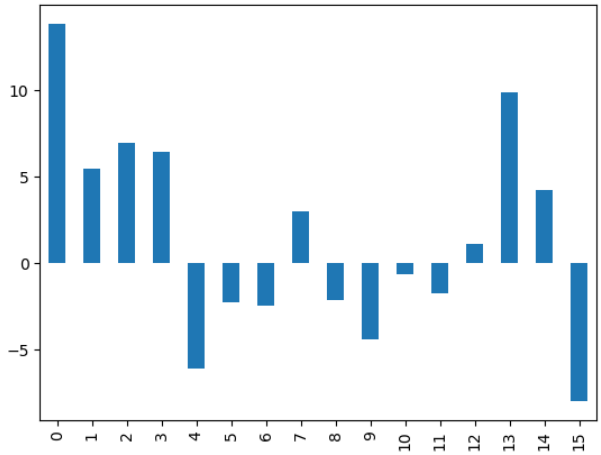
Predicción de los datos de test en SARIMA con ajuste automático de hiperparámetros



Los errores de predicción se muestran en la figura 31.

Figura 31:

Error en la predicción de los datos de test en SARIMA con ajuste automático de hiperparámetros



Las métricas de error para el modelo SARIMA con ajuste automático se muestran en la tabla 10.

Tabla 10:

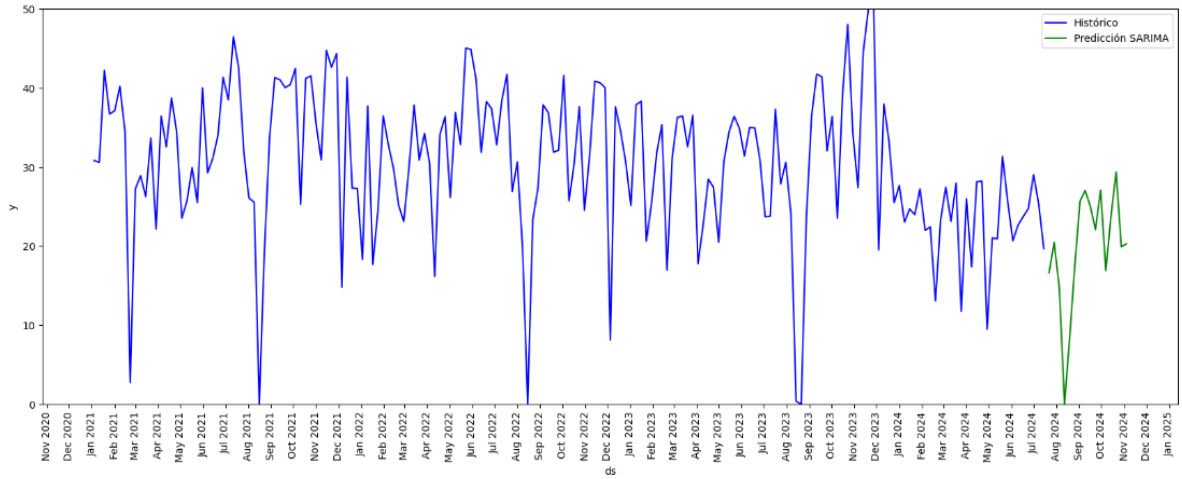
Métricas de error para el modelo SARIMA

	RMSE	MAE	MAPE
SARIMA	6	4,918	22,79%

Por último, la predicción final del modelo SARIMA puede verse en la figura 32. Como puede observarse, el modelo ha sido capaz de predecir la parada de agosto, lo cual indica que ha sido capaz de comprender los patrones subyacentes en la serie.

Figura 32:

Predicción de la carga de planta en las próximas 16 semanas con el modelo SARIMA



5.4.3. Prophet.

Este es el tercer modelo que se va a entrenar para predecir la serie temporal objetivo. Sus fundamentos han sido explicados en la sección 2.3.3.

5.4.3.1. Entrenamiento.

Tras la preparación de los datos explicada en la sección 5.3, se ha definido un calendario con los eventos (cierres) de la serie histórica y del futuro. Estos eventos corresponden a los cierres programados en planta (la mayoría en agosto, debido a la parada anual de mantenimiento) y se muestran en la tabla 11.

Tabla 11:

Fechas de cierre programado para la serie.

ds	holiday
22/02/2021	cierre_programado
16/08/2021	cierre_programado
15/08/2022	cierre_programado
14/08/2023	cierre_programado
21/08/2023	cierre_programado
05/08/2024	cierre_programado
12/08/2024	cierre_programado

Al igual que en modelos anteriores se ha realizado un ajuste iterativo de hiperparámetros, en este caso y según la documentación oficial de Prophet (Meta, 2023) el hiperparámetro más conveniente a ajustar es “changepoint_prior_scale” que controla la capacidad del modelo para adaptarse a cambios en la tendencia. El potencial de ajuste de este hiperparámetro también ha sido ampliamente explorado por otras fuentes (Peak Maximum, 2018), demostrando su valor a hora de ajustar la serie temporal de manera correcta.

Se han probado iterativamente valores entre 0.01 y 3 con un salto de 0.01, dando como resultado que el mínimo RMSE se obtiene con un valor de 3. Sin embargo, este parámetro no ha de ajustarse por encima de 0.5 o, muy excepcionalmente 1, ya que un valor superior da lugar a una gran cantidad de variaciones en la tendencia que tiende a producir un sobreajuste de los datos de entrenamiento, dando lugar a un modelo que no es capaz de generalizar la serie.

El valor de las componentes de tendencia, estacionalidad y cierres de la serie pueden observarse en las figuras 33, 34 y 35 respectivamente.

Figura 33:

Tendencia de la serie temporal según Prophet con "changepoint_prior_scale" optimizado iterativamente

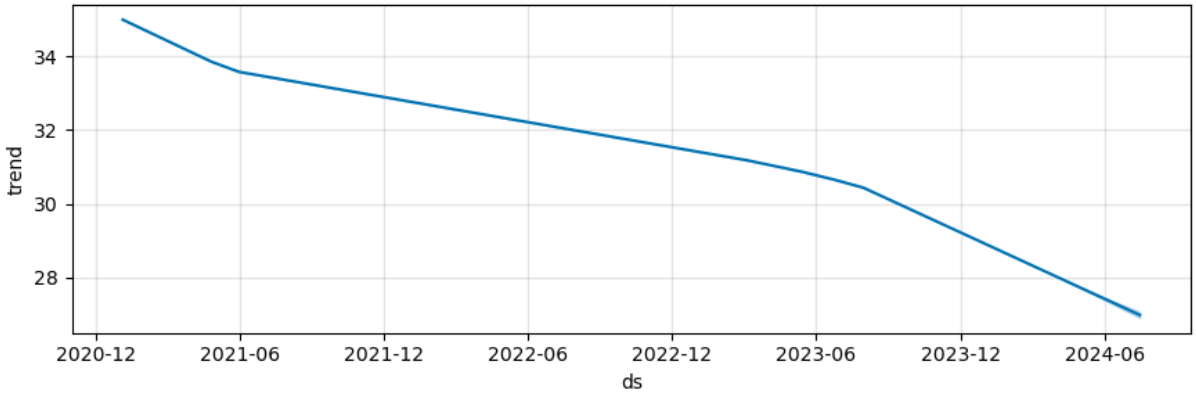


Figura 34:

Estacionalidad de la serie temporal según Prophet

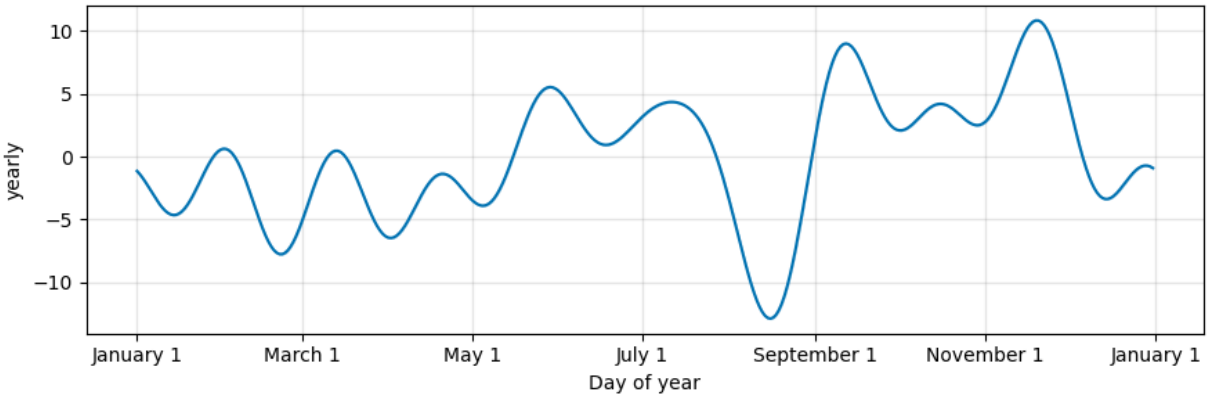
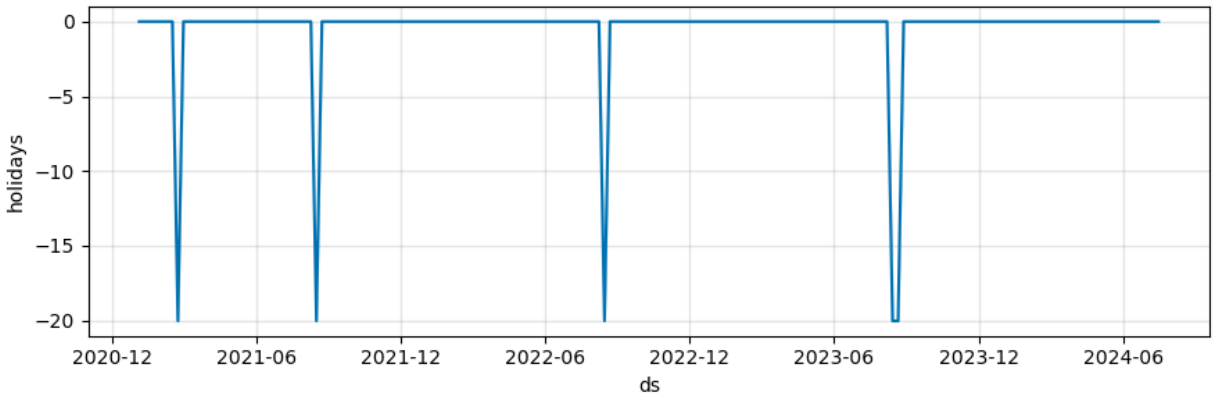


Figura 35:

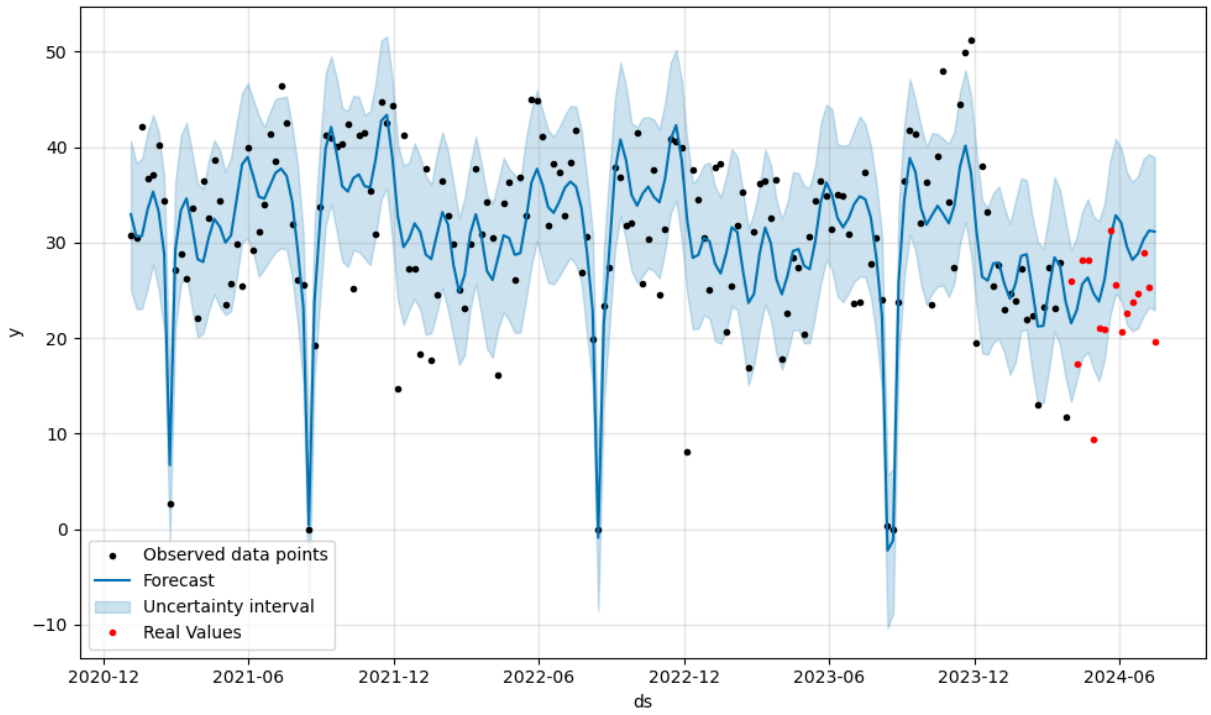
Efectos de los cierres según Prophet.



En la figura 36 puede observarse la predicción del modelo Prophet para los datos de test, el área azul corresponde a un intervalo de confianza del 80%. Es fácil comprender que el modelo ha predicho un aumento de la carga de trabajo que no se ha correspondido con los datos observados.

Figura 36:

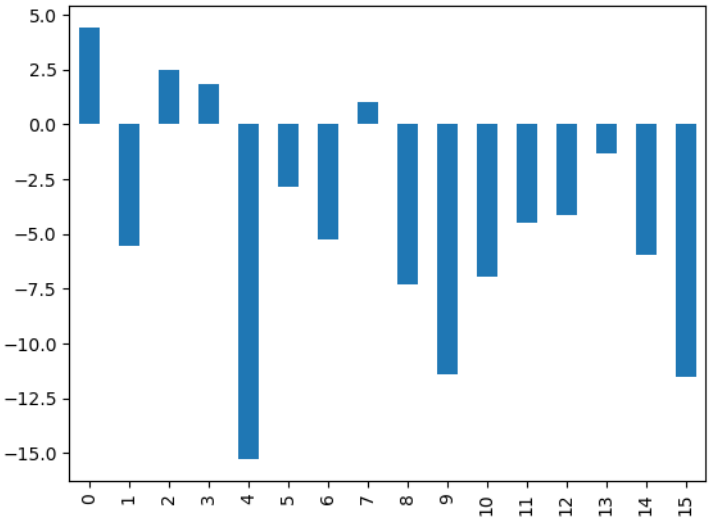
Predicción de los datos de test en Prophet con ajuste automático de hiperparámetros



Esto da lugar a la gráfica de errores mostrada en la figura 37.

Figura 37:

Error en la predicción de los datos de test en Prophet.



Las métricas de error para el modelo PROPHET con ajuste automático se muestran en la tabla 12.

Tabla 12:

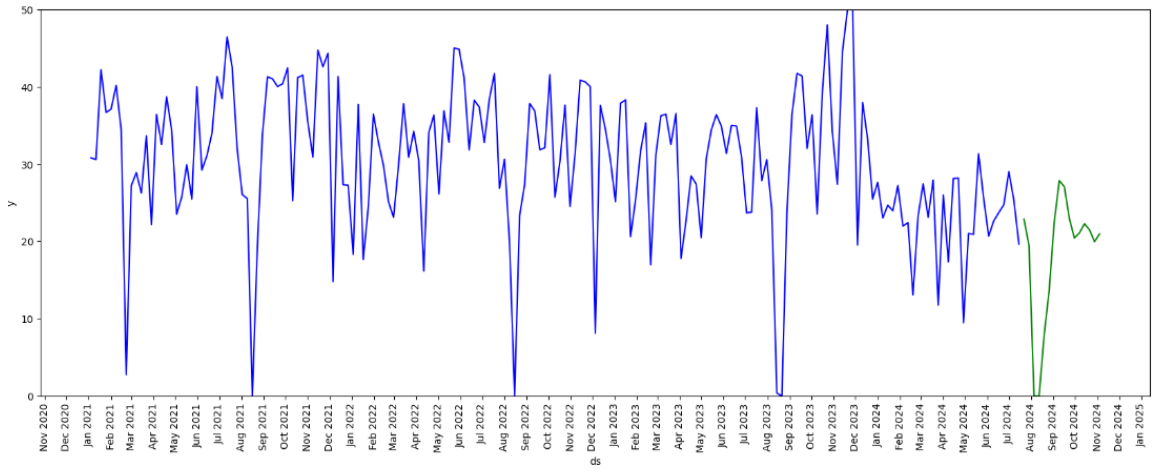
Métricas de error para el modelo Prophet.

	RMSE	MAE	MAPE
PROPHET	6,29	5,73	31,50%

Finamente, la predicción definitiva del modelo puede observarse en la figura 38. Al igual que el resto de los modelos, Prophet ha sido capaz de pronosticar correctamente el cierre de agosto, mostrando una generalización del comportamiento de la serie temporal.

Figura 38:

Predicción de la carga de planta en las próximas 16 semanas con el modelo Prophet.



5.4.4. XGBoost

Este es el cuarto modelo que se va a entrenar para predecir la serie temporal objetivo. Sus fundamentos han sido explicados en la sección 2.3.4.

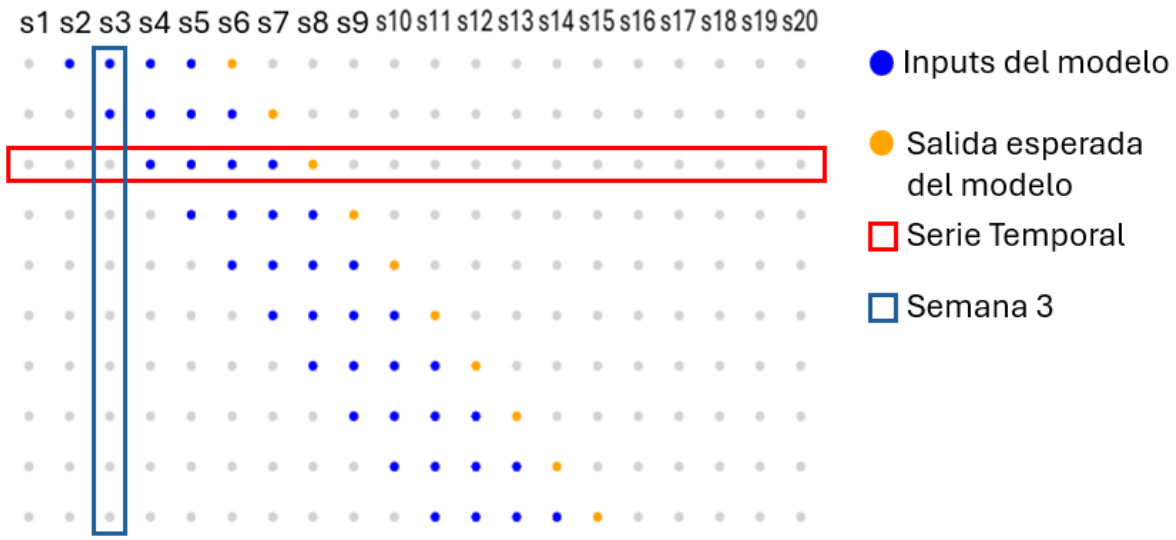
5.4.4.1. Entrenamiento iterativo

Dado que el modelo XGBoost no está diseñado específicamente para la predicción de series temporales, es preciso realizar un pretratamiento de los datos para tratar el problema como si fuese un problema de regresión convencional. Esto significa que, como se detalla en el artículo “Forecasting series temporales con XGBoost” (Amat Rodrigo & Escobar Ortiz, 2024) o el trabajo “How to Use XGBoost for Time Series Forecasting” (Brownlee, 2021), la serie temporal ha de dividirse en un conjunto de inputs con los que pueda alimentarse al modelo y un output que será el valor a estimar.

La figura 39 representa como se ha realizado esta transformación: Cada fila representa la serie temporal, los puntos azules representan los datos con los que se alimenta al modelo y el punto amarillo es la salida esperada del mismo. Se define una cantidad de semanas con las que alimentar al modelo (en la figura se han representado 4 semanas) y se generan datos de entrada y salida para el modelo al correr una ventana con dichas semanas a lo largo de la serie temporal.

Figura 39:

Representación de secuenciación de serie temporal para entrenamiento de XGBoost.



Además, dado que se trata de un modelo de regresión, el modelo permite añadir variables adicionales que puedan ser de utilidad a la hora de realizar predicciones. En este caso, los datos de entrenamiento se prepararán como secuencias de los 52 valores anteriores (s1-s52) a la fecha que se trata de estimar (las 52 semanas previas que componen un año completo) más cuatro columnas representando el año V_1, el mes del año V_2 (de 1 a 12), la semana del año V_3 (de 1 a 52) y una variable booleana V_4 que representa si nos encontramos en una semana de vacaciones o no. Todas las variables que representan valores de carga de trabajo en cada semana s1-s52 se han escalado entre 0 y 1 para evitar descompensaciones entre las mismas. Las variables que representan el año, mes, semana y vacaciones se han dejado sin escalar.

Tabla 13:

Datos de entrenamiento para XGBoost.

	s1	s2	s3	...	s51	s52	V_1	V_2	V_3	V_4	y (salida)
0	0.601595	0.597055	0.824339	...	0.533527	0.532517	2022.0	1.0	1.0	0.0	0.357451
1	0.597055	0.824339	0.716122	...	0.532517	0.357451	2022.0	1.0	2.0	0.0	0.736225
2	0.824339	0.716122	0.724786	...	0.357451	0.736225	2022.0	1.0	3.0	0.0	0.345100
3	0.716122	0.724786	0.784645	...	0.736225	0.345100	2022.0	1.0	4.0	0.0	0.478813
4	0.724786	0.784645	0.672132	...	0.345100	0.478813	2022.0	1.0	5.0	0.0	0.712076
...	...	...	...	...	...	...	...	...	...	...	...
128	0.682044	0.604069	0.462648	...	0.403598	0.441979	2024.0	6.0	25.0	0.0	0.463160
129	0.604069	0.462648	0.464236	...	0.441979	0.463160	2024.0	6.0	26.0	0.0	0.482919
130	0.462648	0.464236	0.728618	...	0.463160	0.482919	2024.0	7.0	27.0	0.0	0.566438
131	0.464236	0.728618	0.543851	...	0.482919	0.566438	2024.0	7.0	28.0	0.0	0.494397
132	0.728618	0.543851	0.596877	...	0.566438	0.494397	2024.0	7.0	29.0	0.0	0.383668

En la tabla 13 pueden observarse los datos de entrenamiento para XGBoost. Cada secuencia de entrenamiento se compone de 56 entradas, de las cuales 52 representan los valores previos de la serie, mientras que las últimas 4 entradas representan el año, el mes, la semana y si se trata de una semana de cierre. Es importante remarcar que la variable objetivo de una secuencia corresponde con el último valor de la secuencia siguiente, como puede observarse en las celdas coloreadas de la tabla 13.

De forma análoga al resto de modelos, se ha realizado un ajuste iterativo de los hiperparámetros mostrados en la tabla 12 optimizando el RMSE y llegando a los siguientes valores:

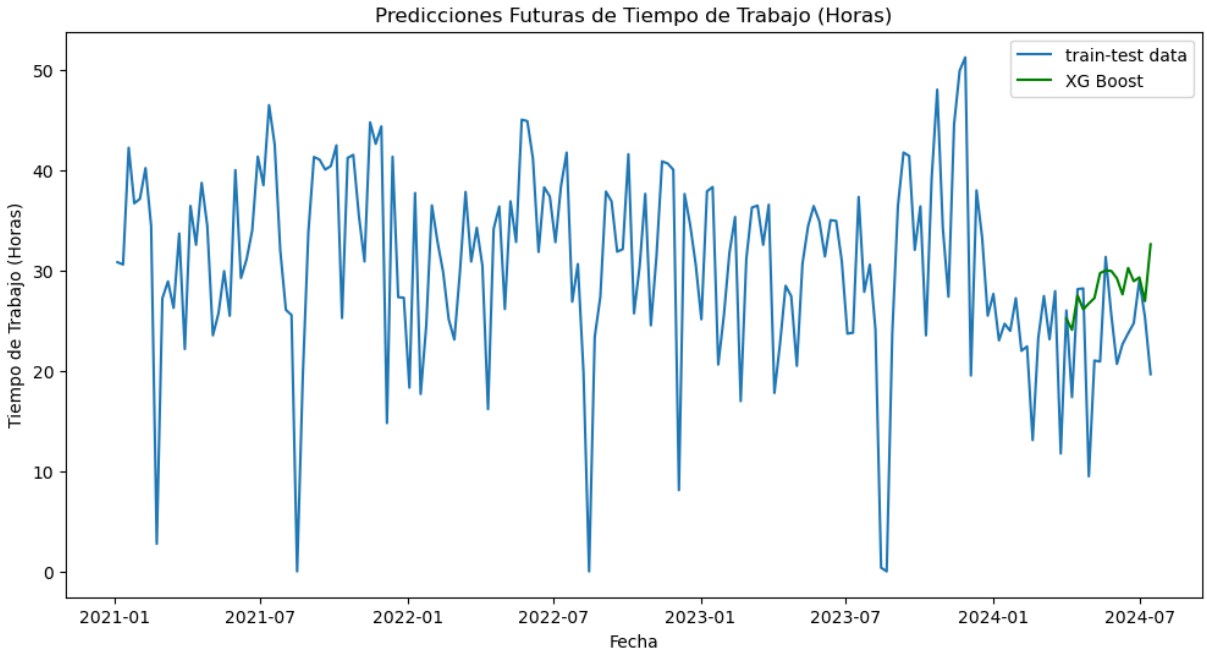
- learning_rate=0.1
- max_depth=10
- subsample=0.7

- colsample_bytree=0.9
- n_estimators=500
- reg_alpha=0
- reg_lambda=1

Sin embargo, una vez entrenado el modelo y realizado el pronóstico para los datos de test, como puede observarse en la figura 40 que existe un sesgo respecto a los valores reales.

Figura 40:

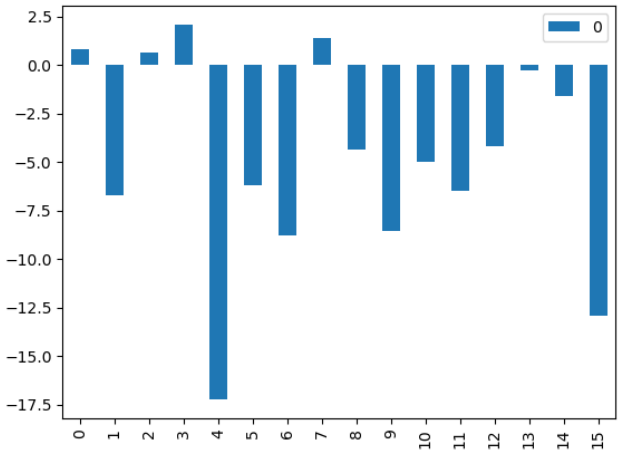
Predicción de los datos de test en XGBoost con ajuste iterativo de hiperparámetros



Como se observa en la figura 41, el modelo tiende a sobreestimar la carga de la planta respecto a los valores reales.

Figura 41:

Error en la predicción de los datos de test en XGBoost

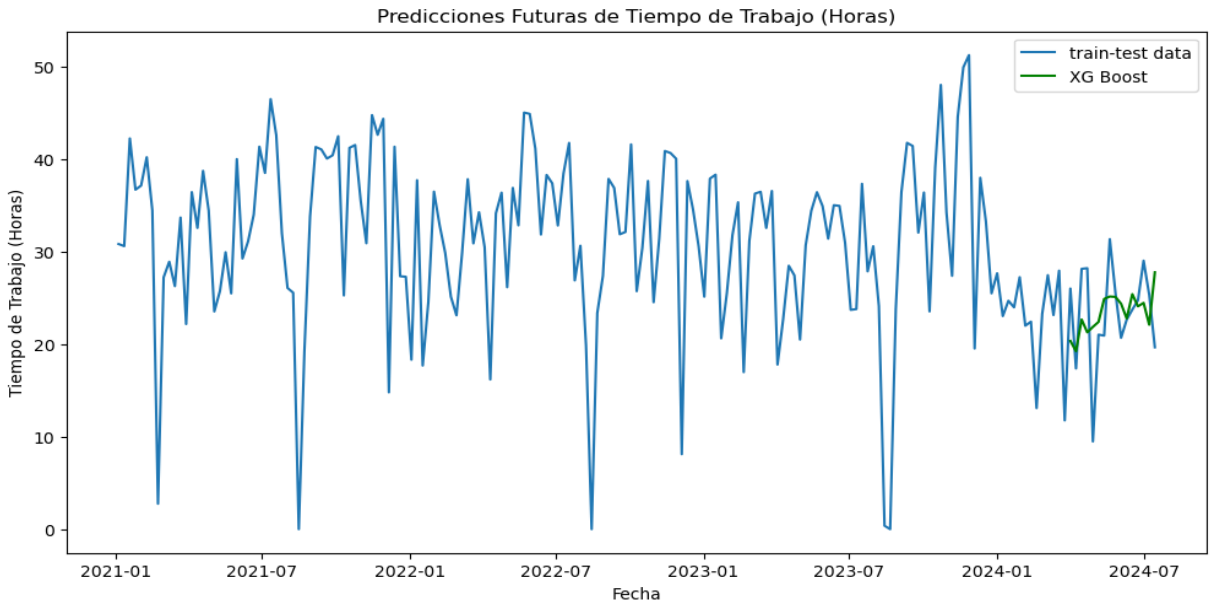


Estas predicciones presentan una desviación media de +4.84 horas con respecto a la realidad. Esto significa que el modelo no ha sido capaz de capturar la complejidad de los datos en su totalidad. Sin embargo, en un modelo de regresión es posible aplicar esa corrección a los datos estimados por el modelo, restando el error medio (Kuhn & Johnson, 2013).

Una vez aplicada la corrección, las predicciones estimadas por el modelo se muestran en la figura 42:

Figura 42:

Predicción de los datos de test en XGBoost con corrección de error.



Las métricas de error para el modelo final de XGBoost se muestran en la tabla 14:

Tabla 14:

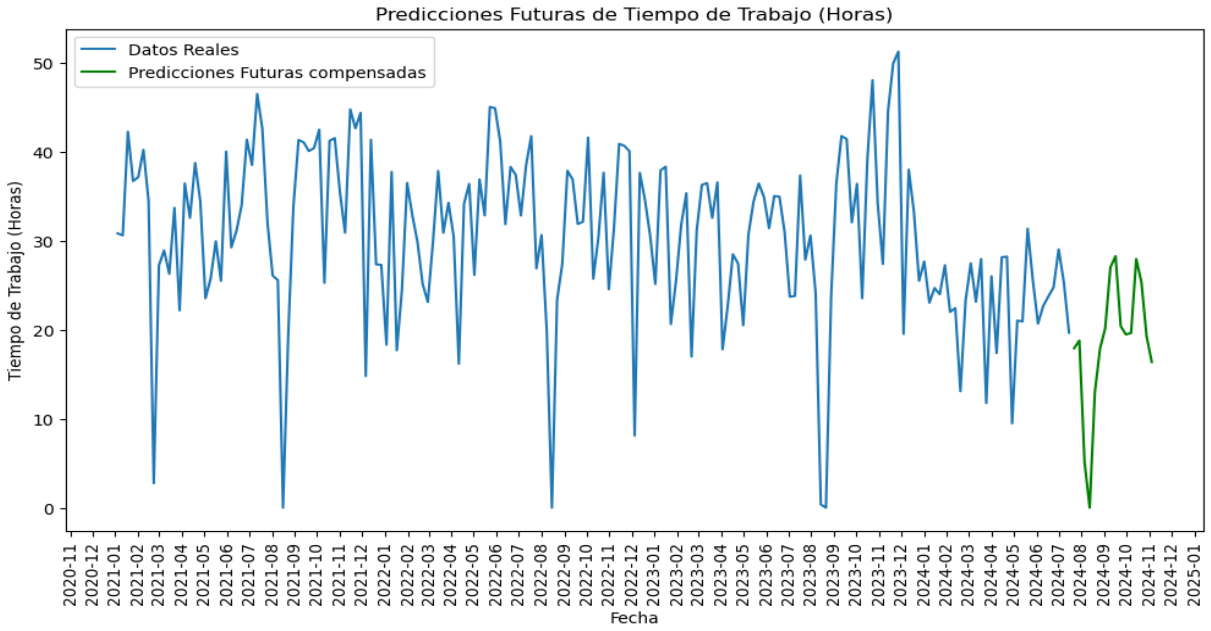
Métricas de error para el modelo XGBoost

	RMSE	MAE	MAPE
XGBoost	5,225	4,15	22,00%

Finamente, la predicción final del modelo puede observarse en la figura 43. El modelo ha sido capaz de pronosticar correctamente el cierre de agosto, mostrando una generalización del comportamiento de la serie temporal.

Figura 43:

Predicción de la carga de planta en las próximas 16 semanas con el modelo XGBoost.



5.4.5. Red Neuronal LSTM

Este es el quinto y último modelo que se va a entrenar para predecir la serie temporal objetivo. Sus fundamentos han sido explicados en la sección 2.3.5.

5.4.5.1. Entrenamiento de la red.

Los datos de entrenamiento para la red neuronal han sido idénticos a los empleados para entrenar el modelo XGBoost mostrados en la tabla 13. Sin embargo, se han eliminado las columnas V_1, V_2, V_3 y V_4 con el objetivo de pasar valores puramente temporales a la red neuronal.

Adicionalmente, como se indica en el trabajo "Deep Learning with Python" (Chollet, 2021), se han escalado los datos de entrenamiento entre 0 y 1 para cada valor de la serie temporal, de esta manera se evita que la escala de los datos afecte al ajuste de la red. Una vez realizada la regresión el resultado se desescala para obtener el valor equivalente en horas de trabajo semanales.

El ajuste de la red se ha realizado probando distintas configuraciones para la misma, siguiendo las prácticas recomendadas en el diseño de redes LSTM (Géron, 2019). Esto ha incluido variaciones en el número de capas y neuronas, así como distintas estrategias para evitar el sobreajuste, tales como la aplicación de regularización L2 y Dropout.

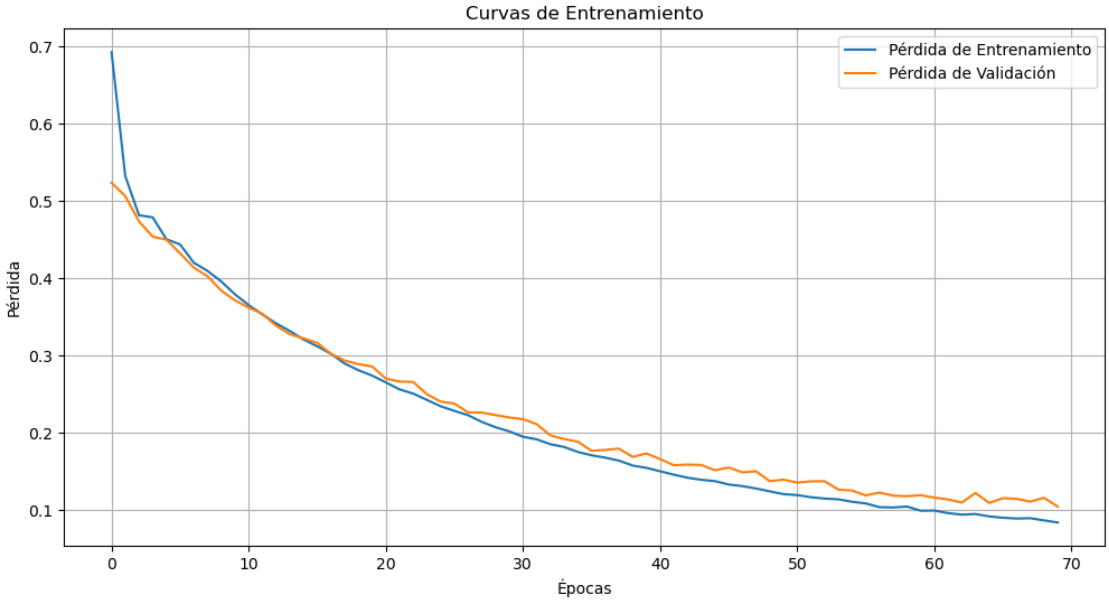
Finalmente, se ha optado por la siguiente estructura de red neuronal:

- Capa de 350 neuronas LSTM, regularizador L2 0.001, dropout de 0.2
- Capa de 150 neuronas LSTM, regularizador L2 0.001, dropout de 0.15
- Capa de 100 neuronas LSTM, dropout de 0.1
- Capa "fully connected" de 1 neurona de salida.
- Tasa de aprendizaje 0.0003.
- Función de pérdida "MSE", Mean Squared Error.
- Early Stopping tras 10 etapas
- 70 Epochs.
- Batch Size 32.

En la figura 44 se puede ver la evolución de la pérdida durante el entrenamiento. Como se puede observar, tanto la pérdida de entrenamiento como la de validación decrecen proporcionalmente y no se muestran signos de sobreajuste por parte del modelo.

Figura 44:

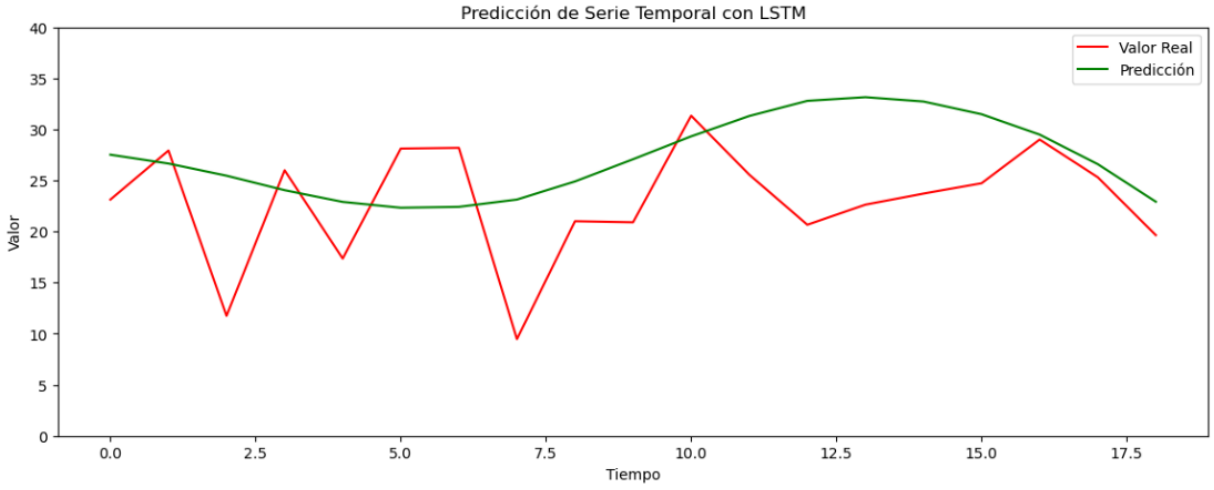
Pérdida de entrenamiento y validación durante el entrenamiento de la red LSTM.



En embargo, al realizar la predicción de los valores de test, en la figura 45 se puede observar que el modelo es capaz de capturar una leve tendencia en los datos, pero no realiza una predicción acertada de los mismos.

Figura 45:

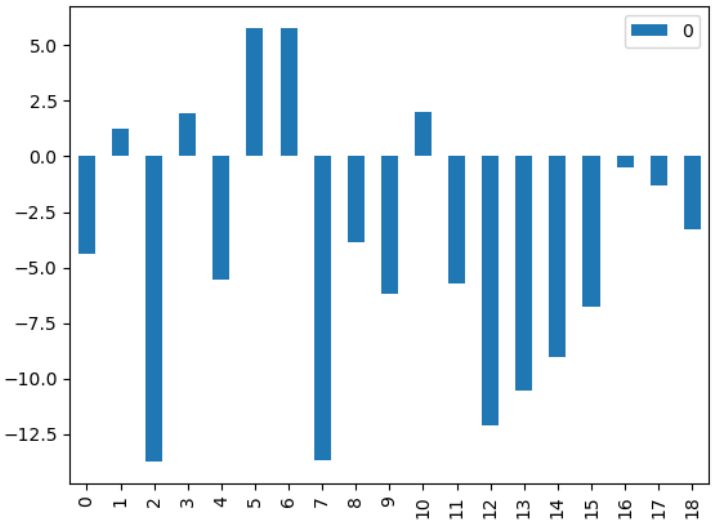
Predicción de los datos de test en red neuronal LSTM (detalle).



Esto da lugar a grandes errores de predicción, como se puede observar en la figura 46.

Figura 46:

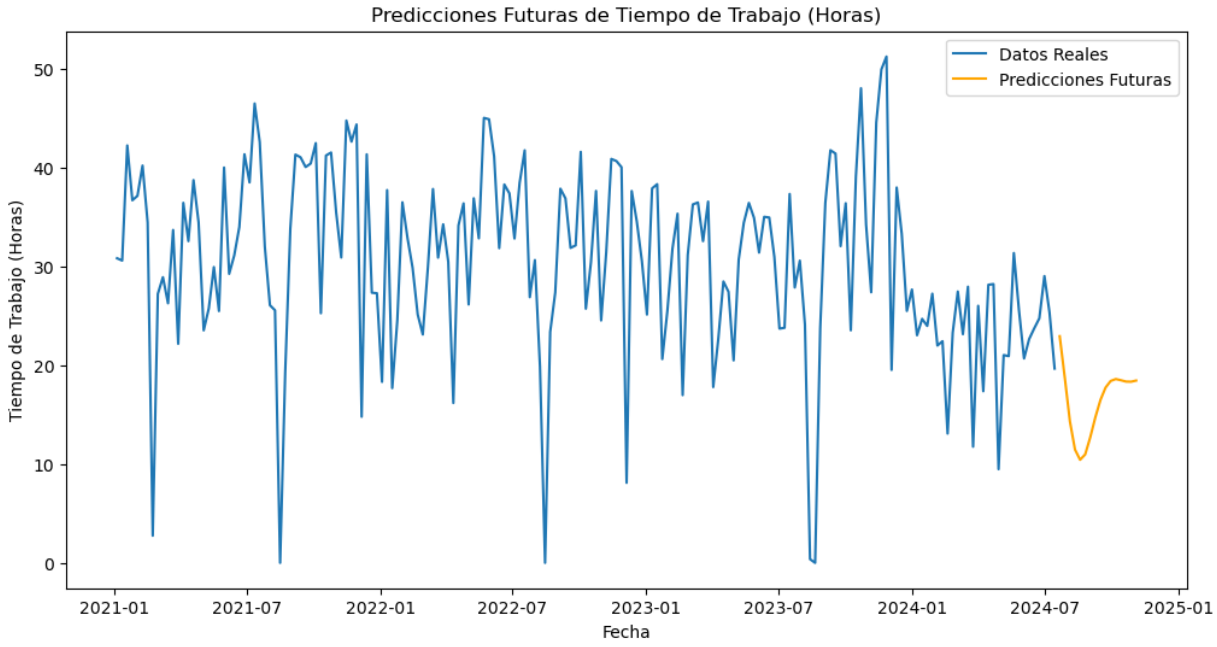
Error en la predicción de los datos de test en red neuronal LSTM



Cuando se realiza la predicción final del modelo, en la figura 47 se observa que existe una tendencia a reducir el volumen durante el cierre de agosto y a incrementarlo al finalizar el mismo. Sin embargo, esta predicción está lejos del nivel de precisión alcanzado por los otros modelos.

Figura 47:

Predicción de la carga de planta en las próximas 16 semanas con el modelo red neuronal LSTM.



La causa de esta falta de rendimiento del modelo puede explicarse en base a la cantidad de datos disponibles. Las redes neuronales necesitan una mayor cantidad de datos de entrenamiento que los modelos clásicos (Brownlee, 2018) y 185 datos de consumos semanales parecen no ser suficientes como para entrenar completamente una red neuronal LSTM.

5.5. Comparación de los modelos.

A lo largo de esta sección, se realiza la comparación final de los distintos modelos predictivos y se evalúa su rendimiento con respecto a los valores reales de la serie para las semanas pronosticadas.

En el apartado 5.4, se han entrenado distintos modelos con el objetivo de predecir 16 semanas de producción. El entrenamiento de estos modelos se ha realizado con los datos disponibles hasta la fecha de entrenamiento del primer modelo (Exponential Smoothing) simulando el proceso real de pronóstico en producción. No obstante, a fecha de escritura de esta sección, ya se dispone de los datos reales de la demanda productiva en la línea y, por tanto, es posible estudiar el desempeño real de los modelos.

Como se puede observar en las figuras 48 y 49, todos los modelos han presentado unas predicciones similares y han sido capaces de predecir correctamente el comportamiento futuro de la serie temporal, sobre todo las 8 primeras semanas. Todos los modelos a excepción de Prophet han predicho correctamente el valor de la serie en la primera semana (22 de julio), no obstante, todos se han quedado cortos en la predicción de la segunda semana (29 de julio). Todos los modelos han sido capaces de predecir el cierre de la cuarta semana (12 de agosto), incluso aquellos a los cuales no se les ha indicado explícitamente, como Exponential Smoothing y SARIMA. Además, todos los modelos han pronosticado correctamente la segunda semana de cierre a excepción de SARIMA, esto tiene sentido ya que a este modelo no se le ha indicado explícitamente que la planta cierra dicha semana.

Todos los modelos han sido capaces de predecir un pico de carga de trabajo del orden de 30 horas para las semanas del 2 y 9 de septiembre, a pesar de que el pico máximo se alcanzó la semana del 2 de septiembre y los modelos lo pronosticaron para el 9. No obstante, a partir de la semana del 16 de septiembre, los modelos Prophet y XGBoost han predicho una caída en la carga de trabajo que no se ha producido, mientras que SARIMA y Exponential Smoothing han acertado la semana del 30 de septiembre. A partir de esa fecha los modelos han pronosticado incorrectamente una caída en la carga de trabajo que, de hecho, se ha incrementado significativamente hasta las 45 horas en la semana del 14 de octubre. Esto se debe a un pedido especial en planta que no estaba contemplado previamente. Finalmente, a partir de la semana del 21 de octubre todos los modelos aciertan sus predicciones hasta la última semana.

Figura 48:

Comparación de la predicción final de distintos modelos vs valores reales.

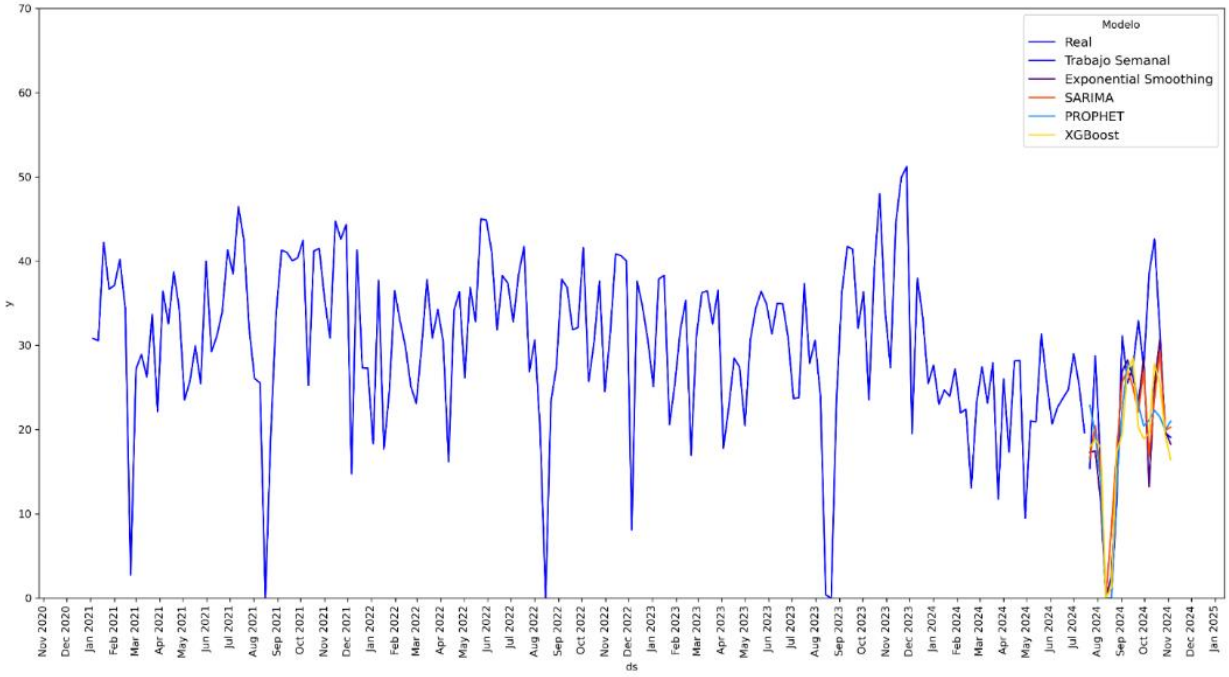
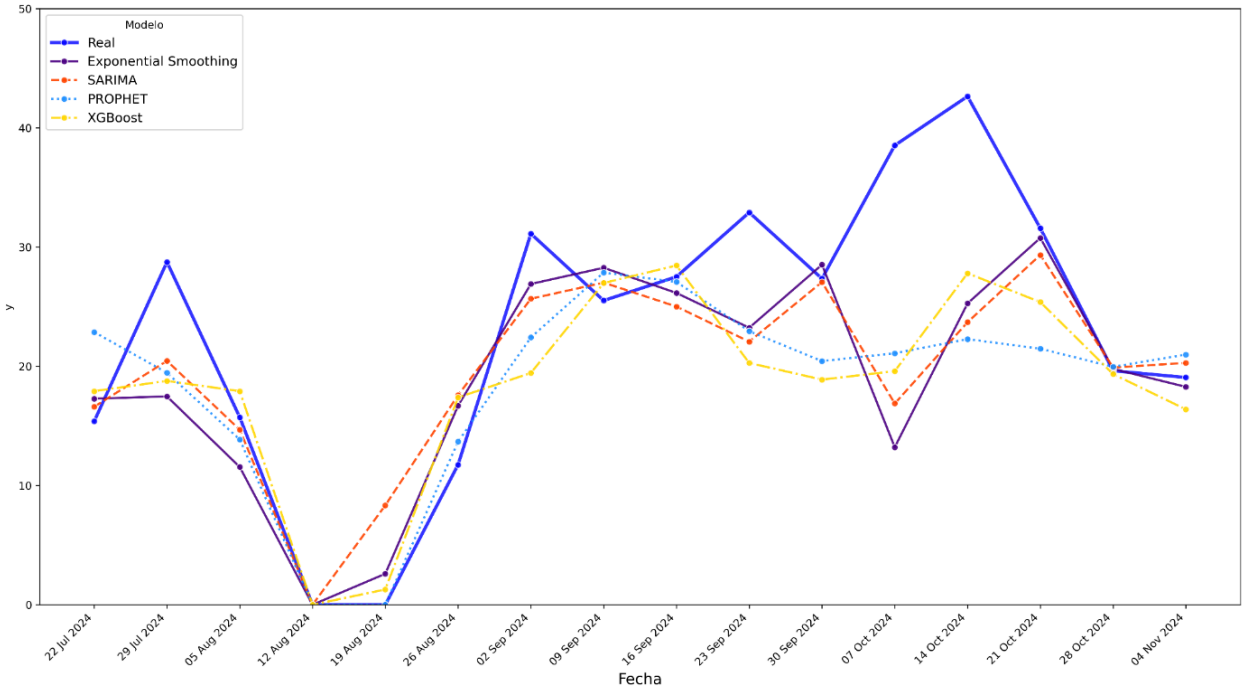


Figura 49:

Comparación de la predicción final de distintos modelos vs valores reales (detalle)



En las figuras 50 y 51 se muestra la evolución del Root Mean Squared Error (RMSE) y Mean Absolute Error (MAE) a lo largo de las semanas para todos los modelos. Para ambas gráficas se comienza con un valor bajo, debido a la acertada predicción inicial que, tras un pico en la segunda semana, se estabiliza en torno a un RMSE de 5 horas y un MAE de 4 horas para todos los modelos excepto XGBoost, que presenta 6 y 4.3 horas respectivamente.

Tras la semana 9 los valores se disparan hasta alcanzar niveles de RMSE de 8.6 horas y MAE de 5.5 horas para Exponential Smoothing y SARIMA y 6.2 horas para Prophet y XGBoost.

Figura 50:

RMSE de la predicción de la carga de planta a 16 semanas vista.

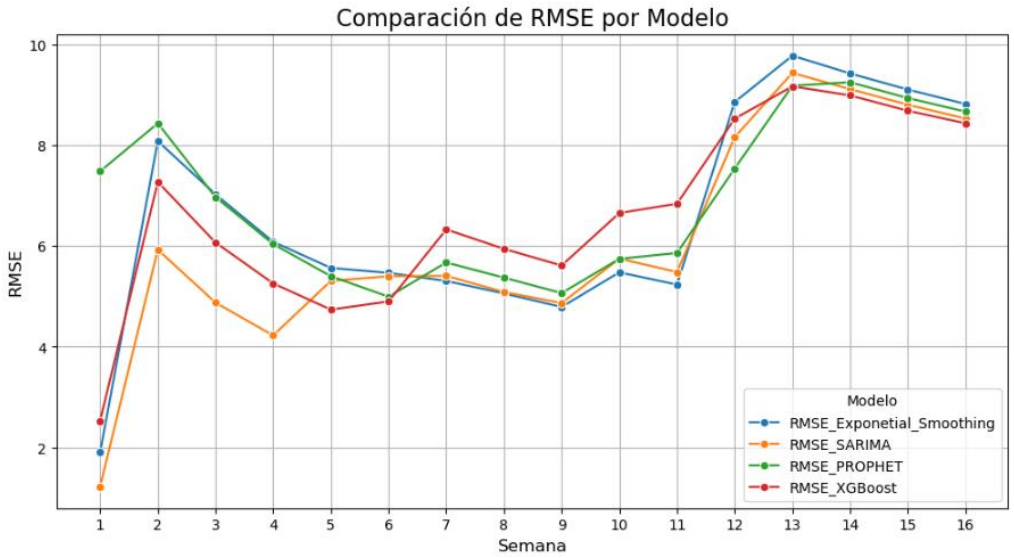
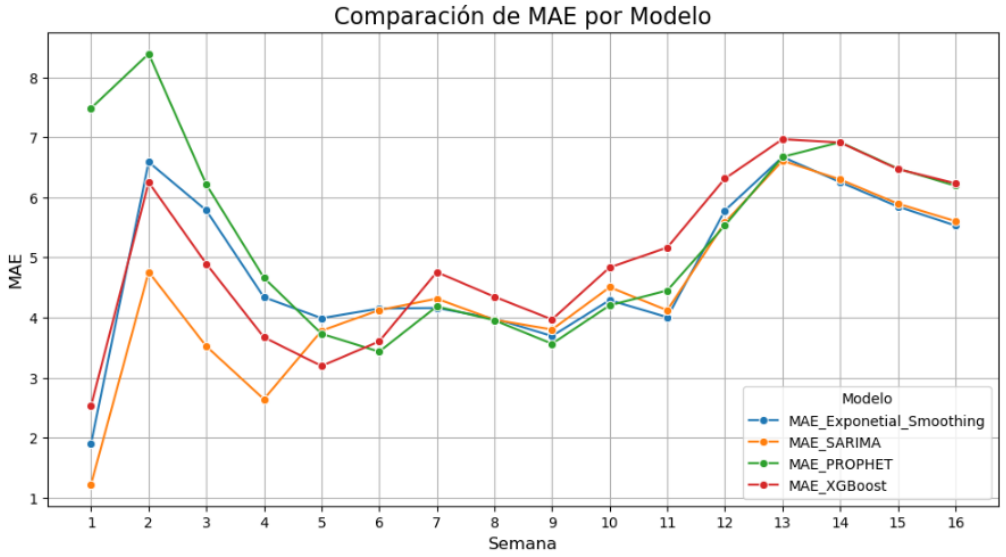


Figura 51:

MAE de la predicción de la carga de planta a 16 semanas vista.



Finalmente, en la tabla 15 se detallan los valores numéricos de las métricas de error mostradas en las figuras 50 y 51.

Tabla 15:

Métricas de error a 8 y 16 semanas para los modelos.

	Previsión 8 semanas		Previsión 16 semanas	
Modelo	**RMSE (horas)**	**MAE (horas)**	**RMSE (horas)**	**MAE (horas)**
Exponential_Smoothing	5,06	3,98	8,82	5,53
SARIMA	5,09	3,96	8,53	5,60
PROPHET	5,37	3,95	8,67	6,19
XGBoost	5,94	4,35	8,44	6,23

La previsión tiene el mismo orden de magnitud en el caso de todos los modelos, no habiendo ninguno que destaque negativamente. Es curioso observar cómo los modelos más complejos (Prophet y XG Boost) han presentado por lo general un RMSE mayor en comparación con los modelos más sencillos a 8 semanas vista, alcanzando valores como 5.37 y 5.94 horas respectivamente frente a las 5.06 y 5.09 horas de Exponential Smoothing y SARIMA.

Esta tendencia no se observa cuando la comparación se hace mediante el MAE para 8 semanas vista ya que todos los valores han sido del orden de 3.96 horas a excepción de XGBoost, que ha presentado 4.35 horas. Esto se debe a que la métrica MAE es menos sensible a los valores extremos que RMSE y, por tanto, si los modelos han sido capaces de capturar la tendencia general de la serie temporal, su valor de MAE se muestra parecido.

Respecto a la previsión a 16 semanas vista, el RMSE ha sido similar en todos los modelos, presentando una ligera mejora el modelo XGBoost (8.44 horas) frente el resto de los modelos, siendo el peor el modelo de Exponential Smoothing (8.82 horas). Es decir, no existe una gran diferencia en el desempeño de los modelos si atendemos al RMSE a 16 semanas.

Respecto al MAE a 16 semanas vista, existen dos órdenes de magnitud claramente diferenciados. Los modelos Exponential Smoothing y SARIMA han presentado 5.53 y 5.6 horas respectivamente mientras que para los modelos Prophet y XGBoost la cifra se eleva a las 6.19 y 6.23 horas.

Cabe decir que un RMSE del orden de 5 horas a 2 meses vista y de 8.5 horas a 4 meses vista representa menos de un turno de trabajo a la semana de error de predicción. Eso permite

calcular los materiales y recursos humanos necesarios para garantizar una producción exitosa, así como reducir stocks intermedios y sobrecapacidad productiva en planta.

5.6.Automatización de la predicción.

Una vez probados y ajustados los modelos y comprendido su alcance, es el momento de automatizar la predicción para que puedan ser empleados en la planta. El objetivo es el de programar una sencilla aplicación que sea capaz de ingerir los datos de entrenamiento y realizar predicciones automáticamente para todas las líneas de fabricación. A lo largo de las siguientes secciones se explica cómo se ha desarrollado la aplicación y cada uno de los programas que la componen.

5.6.1. Backend

Este programa es el encargado de ingerir los datos en crudo procedentes del ERP de la empresa (SAP) y de entrenar los modelos vistos en la sección 5.5. Para ello, se emplea un enfoque similar al de la sección anterior, donde en primer lugar se tratan los datos para generar las series temporales de cada línea de fabricación, a continuación, se realiza la división train-test de cada serie temporal donde los datos de test corresponden a las últimas 16 semanas de cada serie.

Para cada modelo, se realiza un ajuste automático de hiperparámetros usando los datos de train y test y se realiza la predicción de los datos de test con los datos de entrenamiento. El error se evalúa usando dos métricas diferentes: RMSE y MAE, esto permite al usuario comprender de manera sencilla cual es el margen de error de cada modelo en la predicción de datos futuros.

Finamente, se vuelve a realizar un entrenamiento de cada modelo con la serie completa de datos y se genera una predicción de las 16 semanas siguientes. Cada modelo guarda tanto su predicción como su precisión en un dataset propio dentro de la carpeta “data” con formato “.csv”.

5.6.2. Frontend

El frontend se ha programado empleando el framework “Streamlit”, que permite crear aplicaciones web enfocadas a datos de manera sencilla mediante el empleo de funciones predefinidas.

Para ello, en primer lugar se leen los datos a mostrar que contienen las predicciones desde los correspondientes archivos .csv calculados por el backend, a continuación, se muestra un

desplegable que permite elegir la línea de fabricación de la cual quiere visualizarse la predicción.

Como se muestra en la figura 52, para cada línea pueden mostrarse todas las predicciones o solamente un modelo en particular. La selección del modelo se realiza mediante botones.

Figura 52:

Frontend de la aplicación mostrando las predicciones de todos los modelos para la planta completa.

Mientras la predicción de uno o varios modelos se muestra en la gráfica, las métricas de error de cada modelo se exponen en una tabla permitiendo al lector identificar cual es el potencial error en la estimación de cada uno. Por ejemplo, la predicción del modelo exponencial y sus errores para la línea de fabricación de “Paneles” puede observarse en la figura 53.

Figura 53:

Frontend de la aplicación mostrando las predicciones del modelo Exponencial para la línea de Paneles.

Adicionalmente, la gráfica es interactiva, permitiendo hacer zoom a las predicciones para poder comparar cada una de ellas a lo largo de las semanas como se muestra en la figura 54 para la línea de “Tableros”.

Figura 54:

Frontend de la aplicación mostrando las predicciones con zoom de todos los modelos para la línea de Tableros.

Si se requiere realizar una exploración detallada de las predicciones, existe la posibilidad de ampliar la gráfica para ponerla en modo pantalla completa. Además, como se muestra en la figura 55, la gráfica es interactiva, permitiendo explorar los valores de cada predicción pasando el ratón por encima de la línea de la serie temporal correspondiente.

Figura 55:

Detalle de la gráfica interactiva de exploración de predicciones en modo pantalla completa.

6. Código fuente y datos analizados

6.1. Código fuente

En el siguiente link de GitHub, se encuentra el código que se ha empleado para el desarrollo de las secciones 5.3, 5.4 y 5.5 donde, en primer lugar, se realiza una limpieza y pretratamiento del dataset, a continuación, se entrenan diversos modelos para una misma línea productiva y finalmente se realiza una comparación del desempeño de dichos modelos.

El formato empleado para realizar dicho trabajo ha sido Jupyter notebooks, con el objetivo de facilitar la ejecución de bloques de código independientes que permitan desarrollar el código de manera más sencilla.

https://github.com/fjaviermelero/TimeSeriesPrediction

El código de la aplicación descrito en la sección 5.6, que permite realizar la predicción automática de los valores futuros de la serie temporal y mostrarlos a través de una interfaz web se encuentra en el link inferior.

Dado que estos programas han de ser ejecutados por una máquina, tienen extensión ".py". El programa que ejecuta el frontend ha de ser ejecutado a través del framework Streamlit.

https://github.com/fjaviermelero/TimeSeriesApp

6.2. Datos Analizados

Dado que este proyecto se ha realizado en un contexto empresarial, los datos en bruto usados para la realización del mismo no se encuentran disponibles, ya que contienen información sensible acerca de la empresa.

7. Conclusiones

En este proyecto se ha abordado el problema de la planificación eficiente de la capacidad productiva en la industria manufacturera, donde la predicción de la carga de trabajo es fundamental para optimizar los recursos humanos y materiales, evitando ineficiencias como sobrecostes o capacidad ociosa que se producen durante los picos y valles de trabajo. Actualmente la industria del mueble no cuenta con soluciones avanzadas que permitan cuantificar correctamente la carga efectiva de trabajo y predecirla a medio plazo, confiando mayoritariamente en el conocimiento experto para realizar dicha labor.

Para solucionar esta carencia, se ha desarrollado un sistema avanzado basado en modelos de series temporales y técnicas de machine learning. Este sistema permite realizar una cuantificación precisa de la carga de trabajo en cada línea de la fábrica y en la fábrica en su conjunto. El sistema es además capaz de realizar predicciones avanzadas de dicha carga automáticamente a 4 meses vista, mostrando a través de un entorno gráfico intuitivo la predicción de cada modelo, así como sus errores RMSE y MAE.

La solución ha sido validada ya que está fundamentada en los datos reales de producción recogidos en el ERP de la empresa. El desempeño de cada modelo y línea han sido medidos basándose en los datos históricos reales anteriores de cada serie temporal.

Respecto a los objetivos planteados para el trabajo, tanto el objetivo general como los objetivos específicos han sido alcanzados satisfactoriamente.

Objetivo general, desarrollo del sistema:

Dado que ha sido posible desarrollar un sistema predictivo avanzado mediante series temporales que estime la carga de trabajo en cada línea de fabricación para poder planificar de manera eficiente los recursos en planta, puede extraerse la conclusión de que la viabilidad de este sistema queda confirmada para entornos productivos reales.

Objetivo específico 1, procesado de datos para realizar una cuantificación de la carga semanal:

La extracción, limpieza y transformación de los datos ha sido realizada correctamente, generando una representación precisa de la carga de trabajo semanal en planta (medida en horas de trabajo) para cada línea. Se puede concluir por tanto que el sistema de medición de

la carga de trabajo a partir de los escaneos realizados en planta funciona de manera efectiva y es capaz de cuantificar el trabajo correctamente.

Objetivo específico 2, evaluación y selección de modelos predictivos:

Se ha realizado una evaluación de cinco modelos distintos. Los modelos "Exponential Smoothing", "SARIMA", "Prophet" y "XGBoost" han sido capaces de realizar predicciones acertadas y coherentes siendo todas estas similares entre sí. Los modelos han sido capaces de pronosticar correctamente el parón de agosto y han mostrado un RMSE de en torno a 5.5 horas a 8 semanas vista y 8.5 horas a 16 semanas vista para una carga de trabajo de 35 horas, es decir, el margen de error para los modelos se encuentra entre el 15% y el 25%. Por otro lado, el modelo de red neuronal LSTM ha sido capaz de capturar levemente la tendencia de la serie temporal, pero no ha sido capaz de realizar predicciones coherentes debido a la escasez de datos.

De este modo se puede concluir que todos los modelos a excepción de la red neuronal LSTM han sido capaces de realizar predicciones acertadas, capturando correctamente la tendencia y estacionalidad de cada serie. Dentro de estos modelos destacan Exponential Smoothing y SARIMA que, a pesar de ser los modelos más clásicos y sencillos, han demostrado ser también los más robustos.

Objetivo específico 3, automatización de la predicción a todas las líneas y desarrollo de una interfaz gráfica:

El objetivo específico número 3 ha sido alcanzado correctamente ya que ha sido posible automatizar la transformación y limpieza de los datos, para después ajustar automáticamente los hiperparámetros de cada modelo y realizar predicciones en el backend de la aplicación. Por otra parte, el frontend es capaz de mostrar gráficamente los pronósticos al usuario final y permite a este interactuar con ellos, eligiendo el modelo y línea de fabricación que quiere visualizar y viendo el error RMSE y MAE cometido por cada modelo. Se puede concluir, por tanto, que la automatización de la predicción para cada línea y para la fábrica en su conjunto es factible y muestra unos resultados adecuados.

En conclusión y respondiendo a pregunta de investigación, el sistema desarrollado demuestra la viabilidad de la aplicación automática de modelos predictivos de series temporales en

entornos manufactureros reales, logrando predicciones precisas de la carga de trabajo en planta y facilitando la planificación de los medios productivos industriales.

8. Limitaciones y prospectiva

8.1. Limitaciones

Durante el desarrollo de este proyecto se han encontrado diversas limitaciones que han afectado al mismo. A lo largo de esta sección se describen algunas de las principales.

1. Cantidad de datos limitada:

Dada la arquitectura actual del ERP de la empresa, el sistema realiza un almacenamiento de aproximadamente 3 años de los escaneos de productos en planta. Por otro lado, se ha decidido hacer una agrupación semanal de la carga de trabajo en cada línea, ya que una estimación diaria introduciría mucho ruido en el sistema y una estimación mensual sería demasiado generalista como para planificar los recursos de la planta correctamente. Debido a esto, los modelos entrenados solo han contado con datos de los años 2021, 2022, 2023 y 2024 hasta Julio.

La cantidad de datos limitada no ha supuesto un freno al funcionamiento de la mayoría de las modelos, especialmente los modelos clásicos, sin embargo, el modelo de redes neuronales LSTM no ha sido capaz de captar los patrones subyacentes en los mismos y por eso ha sido descartado para su uso en la aplicación final.

2. Tiempo de entrenamiento para el modelo SARIMA:

Como se ha explicado en la sección 5.4.2, el modelo SARIMA cuenta con 6 hiperparámetros ajustables. Para realizar el ajuste automático de estos 6 hiperparámetros se ha empleado la librería auto.arima, que realiza una búsqueda en el espacio de soluciones de los mejores valores para estos. Sin embargo, este proceso es lento, siendo el tiempo de cálculo de 32 minutos. Esto no supone un problema ya que el programa realiza estimaciones semanales y el backend solo es ejecutado una vez a la semana, aun así, esto podría suponer un problema si se pretenden realizar estimaciones a más corto plazo.

Este mismo problema podría aparecer a la hora de emplear modelos predictivos más avanzados como las redes neuronales RNN o LSTM, las cuales requieren un gran tiempo de entrenamiento en comparación con los modelos clásicos.

3. Limitaciones en la generalización para otros contextos:

A pesar de que el sistema ha demostrado un gran desempeño en el contexto de esta empresa, las particularidades de cada empresa y entorno industrial pueden hacer que la solución no funcione de manera adecuada en otros contextos. Es por eso por lo que, si se decide ampliar el alcance de este proyecto a otros contextos industriales, ha de realizarse un estudio detallado del comportamiento de la solución en dicho contexto para evaluar su buen funcionamiento.

8.2. Trabajo futuro

En esta sección se comentan algunas de las perspectivas futuras que podrían seguirse para continuar con el desarrollo de este proyecto.

1. Migración a la nube.

El proyecto está diseñado para ser ejecutado en un servidor local de la planta, sin embargo, es interesante realizar una migración a la nube en el futuro, ya que de esta manera el sistema podrá ser fácilmente accesible por toda la empresa y no solamente por la planta de Madrid.

2. Incorporación de modelos híbridos (Ensemble Learning).

La utilización de los resultados de los modelos actuales para entrenar un metamodelo podría ayudar a incrementar la precisión de la previsión de carga de trabajo. En el ámbito del machine learning a este método se le conoce habitualmente con el nombre de “Ensemble Learning”.

3. Integración de un sistema de gestión de inventarios.

El proyecto podría contar con un sistema que permita gestionar los inventarios disponibles de cada producto y analizar cómo va a evolucionar su stock a lo largo del tiempo, permitiendo prevenir la rotura de stock en productos de largo tiempo de entrega por parte de proveedores.

4. Integración de un sistema de gestión de mano de obra.

De forma análoga, el proyecto podría contar con un sistema que permita evaluar la mano de obra disponible en cada línea de la planta, y evaluar con antelación la necesidad de incrementar o reducir la misma a medio plazo.

5. Escalabilidad a otras industrias.

El sistema puede ser adaptado a diferentes sectores dentro de la industria manufacturera, permitiendo evaluar su aplicabilidad en entornos con características distintas como la industria del automóvil o la agroalimentaria.

Referencias bibliográficas

[1] Amat Rodrigo, J., & Escobar Ortiz, J. (2024). Forecasting series temporales con XGBoost. https://cienciadedatos.net/documentos/py56-forecasting-series-temporales-con-xgboost

[2] Banco Mundial. (2021). Machine Learning para la predicción de series temporales en indicadores de desarrollo. Universidad Industrial de Santander. https://noesis.uis.edu.co/server/api/core/bitstreams/6cfdb4a9-652d-4eec-b653-d30591e6cefd/content

[3] Blanco, I., García, S. J., & Remesal, Á. (2023). *Aprendizaje profundo para series temporales en finanzas: aplicación al factor momentum*. SSRN. https://ssrn.com/abstract=4668800

[4] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time series analysis: Forecasting and control (5th ed.). Wiley.

[5] Brownlee, J. (2021). How to Use XGBoost for Time Series Forecasting. https://machinelearningmastery.com/xgboost-for-time-series-forecasting/

[6] Brownlee, J. (2018). Deep Learning for Time Series Forecasting. Machine Learning Mastery.

[7] Burnham, K. P., & Anderson, D. R. (2002). Model selection and multimodel inference: A practical information-theoretic approach (2nd ed.). Springer.

[8] Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., ... & Riddell, A. (2020). Stan: A probabilistic programming language. Journal of Statistical Software, 76(1), 1–32. https://doi.org/10.18637/jss.v076.i01

[9] Chatfield, C. (2003). The analysis of time series: An introduction (6th ed.). Chapman and Hall/CRC.

[10] Chen, M. (2024). Deep learning-based time series forecasting: Models and applications (Doctoral dissertation). University of Technology Sydney. https://opus.lib.uts.edu.au/handle/10453/168902

[11] Chen, T., & He, T. (2019). XGBoost: Extreme gradient boosting. R package documentation.

[12] Chollet, F. (2021). *Deep Learning with Python* (2ª ed.). Manning Publications.

[13] Crespo Pereira, D. (2013). Modelos de series temporales para simulación de procesos industriales: Aplicación al dimensionamiento y control de sistemas altamente variables (Tesis doctoral). Universidad de La Coruña. https://ruc.udc.es/dspace/bitstream/handle/2183/11511/CrespoPereira_Diego_TD_2013.pdf

[14] de Myttenaere, A., Golden, B., Le Grand, B., & Rossi, F. (2015). Mean absolute percentage error for regression models. Neurocomputing, 192, 38–48. https://doi.org/10.1016/j.neucom.2015.12.114

[15] García, A. (2015). Predicción en el dominio del tiempo: Análisis de series temporales para ingenieros. Editorial Universitat Politècnica de València. https://m.riunet.upv.es/bitstream/handle/10251/72938/IPP-García%20-%20Predicción%20en%20el%20dominio%20del%20tiempo.%20Análisis%20de%20series%20temporales%20para%20ingenieros.pdf

[16] Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2ª ed.). O'Reilly Media.

[17] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

[18] Hewamalage, H., Bergmeir, C., & Bandara, K. (2020). Global models for time series forecasting: A simulation study. arXiv preprint. https://arxiv.org/abs/2012.12485

[19] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

[20] Holt, C. C. (2004). Forecasting trends and seasonals by exponentially weighted moving averages. International Journal of Forecasting, 20(1), 5–10.

[21] Hyndman, R. J., & Athanasopoulos, G. (2021). Forecasting: Principles and practice (3rd ed.). OTexts.

[22] Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for R. Journal of Statistical Software, 27(1), 1–22. [DOI]

[23] Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. International Journal of Forecasting, 22(4), 679–688. https://doi.org/10.1016/j.ijforecast.2006.03.001

[24] ISO (2022). ISO/IEC 27001:2022: Information security, cybersecurity and privacy protection — Information security management systems — Requirements. International Organization for Standardization.

[25] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 30, 3146–3154.

[26] Kovacs, T. (2023). An adaptive learning approach to multivariate time forecasting in industrial processes (Doctoral dissertation). University of Toulouse. https://arxiv.org/abs/2403.07554

[27] Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer. https://doi.org/10.1007/978-1-4614-6849-3

[28] Kumar, A. (2023). *Random Forest vs XGBoost: Which One to Use?* Vitalflux. https://vitalflux.com/random-forest-vs-xgboost-which-one-to-use/

[29] Lago, J., Marcjasz, G., De Schutter, B., & Weron, R. (2021). Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. Applied Energy, 293, 116983. https://doi.org/10.1016/j.apenergy.2021.116983

[30] Li, C., Xiao, P., & Yuan, Q. (2024). FPN-fusion: Enhanced linear complexity time series forecasting model. arXiv preprint. https://arxiv.org/abs/2406.06603

[31] Liker, J. K. (2004). The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer. McGraw-Hill.

[32] Liu, Y. (2024). Multi-variable adversarial time-series forecast model (Doctoral dissertation). Stanford University. https://arxiv.org/abs/2406.00596

[33] López Oriona, Á., Montero Manso, P., & Vilar Fernández, J. A. (2023). Time series clustering based on prediction accuracy of global forecasting models. arXiv preprint. https://arxiv.org/abs/2305.00473

[34] Madrigal Espinosa, S. (2014). Modelos de regresión para el pronóstico de series temporales con estacionalidad creciente. Computación y Sistemas, 18(4), 699–710. https://doi.org/10.13053/cys-18-4-2047

[35] Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 Competition: Results, findings, and conclusions. International Journal of Forecasting, 36(1), 54-74. https://doi.org/10.1016/j.ijforecast.2019.04.014

[36] MathWorks. (n.d.). Long Short-Term Memory Networks. En MathWorks. https://la.mathworks.com/help/deeplearning/ug/long-short-term-memory-networks.html

[37] Matplotlib Developers. (2023). Matplotlib Documentation. https://matplotlib.org/

[38] McKinney, W. (2022). Pandas Documentation. https://pandas.pydata.org/docs/

[39] Meta. (2023). Prophet: Forecasting at scale. https://facebook.github.io/prophet/

[40] OCDE (2013). Guidelines on the Protection of Privacy and Transborder Flows of Personal Data. Organisation for Economic Co-operation and Development.

[41] Ohno, T. (1988). Toyota Production System: Beyond Large-Scale Production. Productivity Press.

[42] Orlando, G., Bufalo, M., & Stoop, R. (2022). Financial markets' deterministic aspects modeled by a low-dimensional equation. Scientific Reports, 12(1), 1–12. https://doi.org/10.1038/s41598-022-05794-0

[43] Ortega, P., & Santamaría, R. (2020). Análisis de series temporales: Estudio de un caso práctico. Universidad Politécnica de Madrid. https://oa.upm.es/70926/

[44] Peak Maximum. (2018). Visualizing Prophet's changepoint prior scale. https://peakmaximum.com/2018/12/27/visualizing-prophet-s-changepoint-prior-scale/

[45] Python Software Foundation. (2023). Python Documentation. https://docs.python.org/

[46] Shumway, R. H., & Stoffer, D. S. (2017). Time series analysis and its applications: With R examples. Springer.

[47] Smith, J. (2024). Large scale hierarchical industrial demand time-series forecasting incorporating sparsity (Doctoral dissertation). Massachusetts Institute of Technology. https://arxiv.org/abs/2407.02657

[48] Statsmodels. (2023). statsmodels.tsa.holtwinters.ExponentialSmoothing.fit. Statsmodels. https://www.statsmodels.org/dev/generated/statsmodels.tsa.holtwinters.ExponentialSmoothing.fit.html

[49] Stevenson, W. J. (2021). Operations Management. McGraw-Hill Education.

[50] Streamlit Inc. (2023). Streamlit Documentation. https://docs.streamlit.io/

[51] Taylor, S. J., & Letham, B. (2017). Forecasting at scale. PeerJ Preprints. https://doi.org/10.7287/peerj.preprints.3190v2

[52] Telefónica y Siemens. (2025, 13 de enero). Telefónica y Siemens impulsarán la digitalización de la industria en España. Cinco Días. https://cincodias.elpais.com/companias/2025-01-13/telefonica-y-siemens-impulsaran-la-digitalizacion-de-la-industria-en-espana.html

[53] Tsay, R. S. (2020). Analysis of financial time series (4th ed.). Wiley.

[54] Vinodh, S. (2021). Lean Manufacturing: Fundamentals, Tools, Approaches, and Industry 4.0. CRC Press.

[55] Wang, Z. (2023). Industrial forecasting with exponentially smoothed recurrent neural networks (Doctoral dissertation). Tsinghua University. https://arxiv.org/abs/2004.04717

[56] Waskom, M. (2022). Seaborn Documentation. https://seaborn.pydata.org/

[57] Wei, W. W. S. (2019). Time series analysis: Univariate and multivariate methods (2nd ed.). Pearson.

[58] 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, 79–82. https://doi.org/10.3354/cr030079

[59] Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. Management Science, 6(3), 324–342.

[60] XGBoost Developers. (2024). XGBoost documentation. XGBoost. https://xgboost.readthedocs.io/en/stable/