

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



Facultad de Ingeniería en Electricidad y Computación

Sistema de recomendación para el abastecimiento de
remolcadores en embarcaciones de hidrocarburos utilizando
técnicas de *Machine Learning*

PROYECTO DE TITULACIÓN

Previo la obtención del Título de:

Magister en Ciencias de Datos

Presentado por:

Paola Elizabeth Galarza Ayala, Ing.

Guayaquil - Ecuador

Año 2025

DEDICATORIA

Esta dedico a Johny mi compañero y apoyo incondicional.

A mis amados hijos André, Gabriel y Dannita; quiero decirles que “*somos imparables*” aunque en ocasiones vayamos lentos... no significa fracaso, sino constancia y esfuerzo para lograr los sueños. Y una vez que los alcancen, recuerden siempre, que:

No somos el centro, no nos pongamos por encima de nadie. La humildad te ancla, la disciplina te construye y la constancia te transforma.

AGRADECIMIENTOS

Mi más sincero agradecimiento a la comunidad científica de ciencia de datos: cienciadedatos.net y PyData; quienes comparten sus experiencias por medio de conferencias y documentan sus investigaciones.

DECLARACIÓN EXPRESA

Yo, Paola Galarza Ayala acuerdo y reconozco que: La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores. El o los estudiantes deberán procurar en cualquier caso de cesión de sus derechos patrimoniales incluir una cláusula en la cesión que proteja la vigencia de la licencia aquí concedida a la ESPOL.

La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, secreto empresarial, derechos patrimoniales de autor sobre software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por mí durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que me corresponda de los beneficios económicos que la ESPOL reciba por la explotación de mi innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique al autor que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Ancón, 15 de agosto del 2025

Paola Galarza Ayala, Ing.

COMITÉ EVALUADOR

MSA. Andrés Wong Blacio

PROFESOR TUTOR

Ph.D. María Isabel Mera Collantes

PROFESOR EVALUADOR

RESUMEN

Se aborda el pronóstico para el abastecimiento de remolcadores en un terminal petrolero marítimo. El objetivo propuesto consistió en realizar un análisis descriptivo y predictivo de la demanda de remolcadores mediante técnicas de *Machine Learning*. Se plantea como hipótesis que los modelos ML pueden ser fiables, así como los modelos estadísticos-econométricos y Deep Learning; la justificación radica en la ausencia de herramientas tecnológicas en la logística portuaria. Para el desarrollo de esta solución se utilizó la metodología KDD (Knowledge Discovery in databases) para un enfoque heurístico e empírico. Se integraron y depuraron datos operativos y meteorológicos; se exploró la estructura temporal (nivel, tendencia, estacionalidad y ruido), y se aplicaron pruebas estadísticas y visuales sobre las características de la serie. Se modeló con árboles de gradiente usando la biblioteca Skforecast en un enfoque univariante y multistep, bajo restricción computacional. Para evitar *data leakage* se recurrió a la validación cruzada en conjunto con la técnica backtesting. Se adaptó métricas apropiadas a la serie para evaluar el desempeño bajo condiciones reales, reforzando la validez en contextos operativos con alta variabilidad e incertidumbre. En los resultados se obtuvo un modelo que alcanzó 81.11% de cobertura y con una precisión de $RMdSE=0.051340$. Como conclusión, esta propuesta sienta el inicio de las bases para una toma de decisiones más informada, eficiente y proactiva en un sector estratégico del país.

Palabras clave: Skforecast; pronóstico de series temporales; remolcadores; puertos petroleros; forecasting.

ABSTRACT

The forecast for the supply of tugboats at a maritime oil terminal is addressed. The proposed objective was to perform a descriptive and predictive analysis of the demand for tugboats using machine learning techniques. The hypothesis is that ML models can be reliable, as can statistical-econometric and deep learning models; the justification lies in the absence of technological tools in port logistics. To develop this solution, the KDD (Knowledge Discovery in Databases) methodology was used for a heuristic and empirical approach. Operational and meteorological data were integrated and refined; the temporal structure (level, trend, seasonality, and noise) was explored, and statistical and visual tests were applied to the characteristics of the series. Gradient trees were modeled using the Skforecast library in a univariate and multistep approach, under computational constraints. To avoid data leakage, cross-validation was used in conjunction with the backtesting technique. Appropriate metrics were adapted to the series to evaluate performance under real conditions, reinforcing validity in operational contexts with high variability and uncertainty. The results obtained a model that achieved 81.11% coverage and an accuracy of $RMdSE=0.051340$. In conclusion, this proposal lays the foundation for more informed, efficient, and proactive decision-making in a sector.

Keywords: *Skforecast; time series forecasting; tugboats; oil ports; forecasting.*

ÍNDICE GENERAL

COMITÉ EVALUADOR	5
RESUMEN.....	6
ABSTRACT	7
ÍNDICE GENERAL	8
ABREVIATURAS.....	11
ÍNDICE DE FIGURAS	12
ÍNDICE DE TABLAS	15
CAPÍTULO 1.....	16
1. PLANTEAMIENTO DE LA PROBLEMÁTICA.....	16
1.1 Descripción del problema	16
1.2 Justificación del problema	18
1.2.1 General.....	18
1.2.2 Específico	19
1.3 Solución propuesta	20
1.4 Objetivos	20
1.4.1 Objetivo General	20
1.4.2 Objetivos Específicos	20
1.5 Metodología	21
1.6 Dataset.....	21
CAPÍTULO 2.....	24
2. dESTADO DE ARTE	24
2.1 Inteligencia Artificial (IA) en la logística marítima	24
2.2 Trabajos de investigación relacionados	25
2.3 Series temporales.....	27
2.3.1 Modelos de predicción en series temporales	28
2.3.2 Forecasting.....	28
2.4 Modelos <i>Machine Learning</i> para <i>Forecasting</i>	31
2.4.1 Introducción a los árboles de decisión.....	31
2.4.2 Árboles de decisión: Extrapolación y Forecasting.....	33

2.4.3	Algoritmos combinados (<i>ensemble</i>)	33
2.5	Técnicas de <i>Gradient Boosting</i> adaptadas a <i>forecasting</i>	35
2.5.1	XGBoost (Extreme Gradient Boosting)	36
2.5.2	LightGBM (Light Gradient Boosting Machine)	36
2.6	Biblioteca SKForecast	36
2.6.1	<i>Forecaster</i> : objeto orquestador y estrategias de predicción	36
2.6.2	Transformación de la serie temporal para <i>forecasting</i>	38
2.6.3	Backtesting	41
2.6.4	Forecasting Probabilístico	44
2.7	Métricas en Forecasting	48
2.7.1	Cálculo de errores y Cálculo de métricas de error: Separación Conceptual en la Evaluación de Modelos Forecasting en Series Temporales	49
2.7.2	Niveles de agregación de error en la evaluación de series temporales	50
2.7.3	Métrica dependiente de la escala: Root Median Squared Error (RMdSE)	52
2.7.4	Métrica basada en porcentajes de error: sMSE (Scaled Mean Squared Error)	52
2.7.5	Combinación de métricas	53
2.8	Selección del modelo, herramienta, técnica y enfoque	54
2.8.1	Selección del modelo acorde a la metodología KDD	55
2.8.2	Selección de la herramienta para el modelado con <i>machine learning</i>	56
2.8.3	Selección de enfoque acorde a la herramienta y capacidad de computo	57
CAPÍTULO 3.....		58
3.	RESULTADOS Y ANÁLISIS	58
3.1	Problema	58
3.2	FASE 1: Selección de datos	58
3.2.1	Recolección	58
3.2.2	Primer cribado	58
3.2.3	Análisis Exploratorio de Datos (EDA): descriptivo	60
3.2.4	Segundo cribado	62
3.3	FASE 2: Pre-procesamiento	64
3.3.1	Identificación del objeto de predicción y reestructuración del <i>dataset</i>	64

3.3.2	Exploración estadística y visual de la estructura temporal de la serie	65
3.4	FASE 3: Transformación para <i>forecasting</i>	73
3.4.1	Transformación a <i>dataset</i> supervisado.....	73
3.4.2	Ingeniería de características (<i>time series feature engineering</i>).....	74
3.5	FASE 4: Modelado y Evaluación.....	77
3.6	FASE 5: Interpretación del conocimiento.....	85
3.6.1	Precisión del modelo top1: RMdSE_2XGB	85
3.6.2	Confiabilidad del modelo top1: RMdSE_2XGB	87
3.6.3	Métrica y distribución del error por paso del modelo top1: RMdSE_2XGB	90
3.6.4	Explicabilidad del modelo top1: RMdSE_2XGB	91
3.6.5	Trazabilidad y composición del modelo top1: RMdSE_2XGB	92
CAPÍTULO 4.....		95
4.	Conclusiones Y Recomendaciones	95
BIBLIOGRAFÍA		98

ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
SUINLI	Superintendencia del Terminal Petrolero La Libertad
INAMHI	Instituto Nacional de Meteorología e Hidrología
SEDC	Sistema de Estandarización de Datos Hidroclimáticos Crudos
NOAA	National Oceanic and Atmospheric Administration
DWD	Deutscher Wetterdienst
OMM	Organización Meteorológica Mundial
IA	Inteligencia Artificial
AAAI	Asociación para el Avance de la Inteligencia Artificial
ML	Machine Learning
DNN	Deep Neural Network
AIS	Automatic Identification System
VHF	Very High Frequency
KDD	Knowledge Discovery in databases
LR	Regresión Lineal
ARIMA	Media Móvil Integrada Autorregresiva
OR	Investigación operativa
MIS	Sistema de Información de Gestión portuaria

ÍNDICE DE FIGURAS

Ilustración 1. Ubicación de dársenas en la zona portuaria de la Superintendencia Terminal Petrolero de La Libertad. Izquierda: 6 dársenas. Centro: 1 dársena superior izquierda. Derecha: Monteverde 1 dársena superior derecha. Fuente: Reglamento de Operaciones de SUINLI	17
Ilustración 2. Componentes del proceso KDD. Fuente: Fayyad <i>et al.</i> , 1996	21
Ilustración 3. Vista panorámica de la analítica de negocios en las operaciones portuarias. Fuente: Filom <i>et al.</i> , 2022.....	25
Ilustración 4. Ejemplo de uniSerie y multiSerie	27
Ilustración 5. Contexto sobre Forecasting Model. Fuente: Abhishek Murthy.....	28
Ilustración 6. Componentes de una serie temporal	29
Ilustración 7. Árboles de Decisión Clasificación. Fuente: https://fhernanb.github.io/libro_mod_pred/arb-de-regre.html	32
Ilustración 8. Árboles de Decisión Regresión. Fuente: https://www.cristobal-aguirre.com/arboles-de-decision	32
Ilustración 9. Extrapolación con modelos de árboles y modelos lineal. Fuente: https://cienciadedatos.net/documentos/py07_arboles_decision_python	32
Ilustración 10. Error total de un modelo en bias + variance + ϵ . Fuente: Link	33
Ilustración 11. Algoritmos <i>ensemble</i> : Bagging & Boosting	34
Ilustración 12. Proceso de los algoritmos <i>Gradient Boosting</i> . Fuente: Link.....	35
Ilustración 13. Estrategia Recursive. Fuente: https://skforecast.org/0.16.0/introduction-forecasting/introduction-forecasting.html	38
Ilustración 14. Estrategia Direct. Fuente: https://skforecast.org/0.16.0/introduction-forecasting/introduction-forecasting.html	38
Ilustración 15. Conversión de un <i>dataset</i> NO supervisado a supervisado. Fuente: Adaptado de Skforecast	40
Ilustración 16. Transformación de series incluyendo una variable exógena. Fuente: Skforecast.....	40
Ilustración 17. Visualización de los Lags y Window Features	41
Ilustración 18. Visualización de las diferencias y transformaciones a una serie temporal.....	41
Ilustración 19. Cross-Validation: K-fold Vs. Backtesting. Fuente: Rokoss <i>et al.</i> , 2024	42
Ilustración 20. Ilustración de los parámetros de Backtesting de la Tabla 6	44
Ilustración 21. Forecasting Probabilístico. Fuente:Skforecast	45
Ilustración 22. Boostrapped residuales. Fuente: Skforecast.....	45
Ilustración 23. Intervalos de predicción con diferentes residuales. Fuente: cienciadedatos.....	46
Ilustración 24. Antes y después del transformador ConformalIntervalCalibrator. Fuente: Skforecast	47
Ilustración 25. Características frecuentes que puede dificultar el pronóstico. Fuente: Kishan Manani	48
Ilustración 26. Error Vs. Métrica. Fuente: Kishan Manani.....	50
Ilustración 27. KDD aplicado al tema de investigación.....	55
Ilustración 28. Tipos de enfoques en <i>Forecasting</i>	57

Ilustración 29. Fiabilidad Vs. Tiempo y memoria. Fuente: Adaptado de Kishan Manani	57
Ilustración 30. Suma de maniobras por: frecuencia diaria resumida desglosada por año	60
Ilustración 31. Suma de maniobras: frecuencia diaria resumida	61
Ilustración 32. Cantidad de Maniobras con Remolcadores: frecuencia diaria en los 5 primeros meses del 2024	61
Ilustración 33. Maniobras de atraque: Número de remolcadores	61
Ilustración 34. Maniobras de desatraque: Número de remolcadores	62
Ilustración 35. Características de buques que arriban al puerto	62
Ilustración 36. Tiempo de estadía por tipo de buque	63
Ilustración 37. Tiempo de servicio de remolcadores con buques por tipo de maniobra y por eslora (longitud) del buque	63
Ilustración 38. Restructuración del <i>dataset</i> para la variable objetivo	65
Ilustración 39. Demanda de remolcadores promedio en una frecuencia de 10 minutos	66
Ilustración 40. Demanda de remolcadores máximo en una frecuencia de 10 minutos	66
Ilustración 41. Prueba visual QQ-plot	67
Ilustración 42. Estacionalidad de la serie temporal	68
Ilustración 43. Autocorrelación de la variable objetivo para 144 pasos (1 día)	68
Ilustración 44. Autocorrelación de la variable objetivo para 1008 pasos (1 semana)	68
Ilustración 45. Detección de outliers combinando las técnicas STL + IQR	70
Ilustración 46. Tendencia, estacionalidad y residuos <i>dataset</i>	70
Ilustración 47. Evolución temporal de los residuos calculados a partir de media móvil simple	71
Ilustración 48. Detección visual de la tendencia con LOESS smoothing	71
Ilustración 49. Distribución de frecuencia de la variable <i>Demand</i>	72
Ilustración 50. Variables exógenas por medio de técnicas <i>feature engineering</i>	75
Ilustración 51. Correlación de Pearson & Permutation Importance	76
Ilustración 52. El resultado RFECV en variables exógenas	77
Ilustración 53. Top en base la métrica RMdSE	78
Ilustración 54. Top en base la métrica sMSE	79
Ilustración 55. Top en base la métrica sMSE+RMdSE	80
Ilustración 56. Primer modelo de la métrica RMdSE	81
Ilustración 57. Forecasting Probabilístico a los modelos seleccionados.	82
Ilustración 58. Cobertura vs. área para las estrategias de Forecasting Probabilístico	82
Ilustración 59. Diagrama de dispersión de la eficiencia de métodos probabilístico en modelos seleccionados	83
Ilustración 60. Métodos probabilísticos que alcanzaron la cobertura calculada $\geq 80\%$	83
Ilustración 61. Top 10 eficiencias filtrado por Cobertura $\geq 80\%$	84
Ilustración 62. Top Final de los mejores modelos acorde a la precisión y confiabilidad	84
Ilustración 63. Mejor modelo: predicciones vs. valores reales	85
Ilustración 64. Diagrama de flujo para la selección de la medida del error de previsión. Fuente: Hewamalage <i>et al.</i> , 2023	86
Ilustración 65. Mejor modelo: intervalo de predicción 80% con la técnica Bootstrapping residuales Out-of-sample (no condicionados)	88

Ilustración 66. Mejor modelo: Residuales off-of-sample.....	89
Ilustración 67. Mejor modelo: métrica del Error por paso	90
Ilustración 68. Mejor modelo: distribución del Error por paso.....	90
Ilustración 69. Explicabilidad del modelo con SHAP Summary	92
Ilustración 70. Trazabilidad de los modelos explorados para esta investigación.....	92

ÍNDICE DE TABLAS

Tabla 1. Tabla arribo de buques con registros desde 1999.....	22
Tabla 2. Tabla remolcadores con registros desde 2016.....	22
Tabla 3. Características de los remolcadores	23
Tabla 4. Diferencias entre árboles tradicionales y árboles forecasting.....	33
Tabla 5. Tipos de Forecaster y sus características para modelos <i>gradient boosting</i>	37
Tabla 6. Forecaster: Recursive Vs. Direct.....	38
Tabla 7. Parámetros de Backtesting	42
Tabla 8. Error Calculation Vs. Error Measure Calculation	50
Tabla 9. Métricas apropiadas a este caso de estudio, basado en Hewamalage <i>et al.</i> (2023).....	54
Tabla 10. Métricas excluidas a este caso de estudio, basado en Hewamalage <i>et al.</i> (2023).....	54
Tabla 11. Dataset unificado entre buque y remolcadores	59
Tabla 12. Registros de remolcadores en la instancia de un buque	59
Tabla 13. Primer cribado del <i>dataset</i>	59
Tabla 14. Interpretación de las gráficas errores por paso	91

CAPÍTULO 1

1. PLANTEAMIENTO DE LA PROBLEMÁTICA

1.1 Descripción del problema

Los puertos marítimos son la sístole de la transportación del comercio mundial a través de rutas en diferentes tipos de buques. Por consiguiente, las operaciones portuarias marítimas conllevan una gran cantidad de datos que pueden ser aprovechados en la industria en conjunto con la academia teniendo por finalidad construir puertos eficientes.

La Superintendencia Petrolera del Ecuador es parte del sector estratégico del estado, funge como autoridad portuaria, gestiona el tráfico marítimo de hidrocarburos a nivel nacional e internacional. Supervisa y controla actividades de maniobra en la transferencia de hidrocarburos a través de buques tanques, mediante cumplimiento de normativas y acuerdos internacionales en materia de prevención de riesgos ambientales.

Los buques de pasajeros, hidrocarburos, portacontenedores y otros vehículos marítimos suelen estar equipados con equipos de comunicación AIS (Automatic Identification System) y VHF (Very High Frequency). El tráfico marítimo en zonas portuarias es difícil de gestionar, especialmente en áreas con gran volumen de buques y una infraestructura de tecnológica insuficientemente desarrollada para satisfacer la demanda.

La Superintendencia del Terminal Petrolero de La Libertad es la más grande de Ecuador -hasta la fecha-, tiene cinco fondeos para líquidos en el cantón La Libertad y un fondeo para gas en la comuna Monteverde (Figura 1). En la logística para las operaciones de traspaso de hidrocarburos se necesita lanchas, remolcadores y prácticos, estos son de carácter privado sin relación de dependencia. Además, carece de soluciones en gestión tecnológica de la información que permita identificar características, patrones y previsiones en las operaciones portuarias a través de software. Las compañías de hidrocarburos propietaria de los buques tienen obligaciones tarifarias para las operaciones marítimas en los puertos. En el Terminal, existen tarifas desde la llegada hasta la partida del buque, incluyendo el tiempo de espera cuando no existe disponibilidad para las operaciones de traspaso de hidrocarburos.

El abastecimiento logístico para la demanda estocástica de buques tiene relevancia por la disponibilidad del remolcador en tipo y cantidad, supeditado al buque por atracar en las dársenas. Por lo general, el arribo de los buques no tiene previo aviso, siendo escasos tales notificaciones al puerto, en el mejor de los casos con 24 o 48 horas.

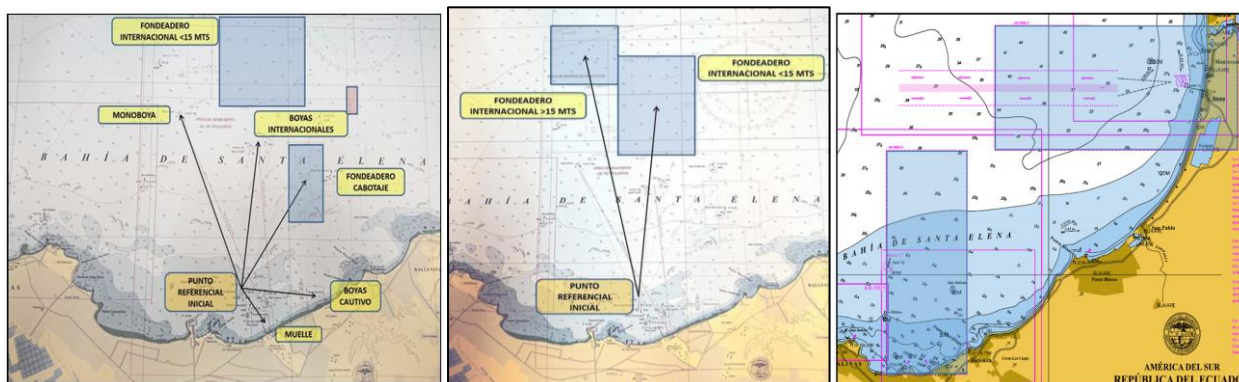


Ilustración 1. Ubicación de dársenas en la zona portuaria de la Superintendencia Terminal Petrolero de La Libertad. Izquierda: 6 dársenas. Centro: 1 dársena superior izquierda. Derecha: Monteverde 1 dársena superior derecha. **Fuente:** Reglamento de Operaciones de SUINLI

Por otra parte, debido a la literatura científica en inglés y en español, existen confusiones en algunos términos, por lo que es necesario aclarar las siguientes definiciones:

- Un “buque” o “ship” se entiende un tanquero de hidrocarburos, un carguero o también barco de pasajeros.
- Una “embarcación” o “vessel” se entiende por un “remolcador” o “tugboat” que asiste en las operaciones de atracadero de los buques.
- La palabra “fondear” es la acción de maniobrar, consiste en estacionar y amarrar el barco al lecho marino mediante una cabo o cadena y utilizando un ancla.
- “Atracar” es la acción de arrimar, acercar un buque a otro, o a tierra.
- “Dársena” es un lugar del puerto resguardada y adecuada para estacionar el barco para las operaciones de carga y descarga de los hidrocarburos. Llamado también, atracadero, desembarcadero, fondeadero, amarradero, o muelle.
- Un “práctico” es un marino experimentado y especializado en el manejo de buques, las particularidades de las aguas y dirección náutica dentro del puerto. Su función es asistir al capitán en tareas de navegación y maniobra en el ingreso y salida del puerto. Aunque, el práctico actúa como asesor en la dirección náutica del buque dentro del puerto, su presencia no disminuye en ningún momento la autoridad del capitán sobre la nave. En los puertos la profundidad disminuye y los espacios son reducidos, por lo que el capitán del buque necesita de la asistencia de “el práctico” entregándole el timón del buque.

1.2 Justificación del problema

1.2.1 General

El comercio marítimo de hidrocarburos es una actividad inherente para la energía y productividad de un país, esto implica numerosas actividades secuenciales y paralelas con respecto a la transportación. Considerando que la mayoría de los principales puertos del mundo están rodeados geográficamente de ciudades, lo que limita su expansión física, los puertos se ven obligados a aumentar su eficiencia en la disminución del costo general de la logística (Wu y Goh, citado por Siyavash, 2022). El uso o desarrollo de herramientas de gestión tecnológica de la información puede mejorar significativamente la eficiencia en la logística de suministro, transporte y distribución en los terminales portuarios (Siddiqui y Rodricks, 2010).

En la segunda década de este siglo se ha desarrollado la digitalización y automatización de los puertos para mejorar la productividad y eficiencia operativa, esta transformación denominada “Smart Port” acuña “Business Analytics” para la calidad de decisiones en las operaciones e inversiones portuarias. Existen varias definiciones para el término Business Analytics proporcionada por Davenport y Harris como “El uso extensivo de datos, análisis estadístico y cuantitativo, modelos explicativos y predictivos y gestión basada en hechos para impulsar decisiones y acciones” (Filom *et al.*, 2022).

Según Filom *et al.* (2022), las aplicaciones de ML en operaciones terrestres son diversas debido a la amplia variedad de maniobras, donde son necesarios el análisis predictivo, el análisis prescriptivo y las automatizaciones. El autor señala que el 68% de los artículos se centraron en el análisis predictivo, mientras que solo el 22% realizaron análisis prescriptivos. En la mayoría de las operaciones portuarias, proporciona una mejor visión basada en el análisis predictivo; sin embargo, se necesita más investigación para cerrar la brecha entre percepción y acción, conocida como análisis prescriptivo (acción), que aporta mayor valor al negocio en comparación con el análisis predictivo (percepción).

En relación con lo expuesto, cabe resaltar que existen varias decisiones críticas que deben resolverse para gestionar los recursos limitados del puerto, por ejemplo, asignación de atracaderos, despacho de buques, carga de combustible, planificación de remolcadores, entre otros. Aunque las operaciones costeras son probablemente el área de operaciones más rica en términos de disponibilidad de datos, el potencial final no se investiga completamente (Filom *et al.*, 2022).

1.2.2 Específico

En Ecuador, el petróleo es el producto de materia prima de mayor exportación, sumado a la importación de sus derivados, por lo que, es primordial la seguridad y eficiencia de las operaciones de los tres Terminales Petroleros Portuarios del país, ubicados en La Libertad, Balao y en Tres Bocas (Bucheli, 2014).

Las operaciones como carga/descarga, almacenamiento y transporte son actividades centrales en los puertos. Cuando un buque arriba al puerto comunica su llegada y los servicios que necesita; las operaciones comienzan desde el arribo, esperar para que un práctico aborde el buque para el control del mismo, este toma las decisiones como el número y características de los remolcadores necesarios, luego con remolcadores se procede atracar el buque en las dársenas adecuadas, seguido se procede al traspaso de hidrocarburos a otro buque o a los tanques en tierra en la refinería por medio conductos submarinos, luego nuevamente con remolcadores se realiza el desatraque de las dársenas dirigiendo al buque a las vías marítimas, y, por último, el práctico entrega al capitán del buque el control para continuar con su ruta de navegación.

El abastecimiento logístico para la demanda estocástica de buques tiene relevancia por la disponibilidad del remolcador en tipo y cantidad, supeditado al buque por atracar en las dársenas. Por lo general, el arribo de los buques no tiene previo aviso, siendo escasos tales notificaciones al puerto, en el mejor de los casos con 24 o 48 horas.

Una herramienta de gestión de la información, enfocado a abastecer la demanda de las operaciones portuarias marítimas en el Terminal Petrolero de La Libertad mejoraría los servicios, partiendo del hecho que no disponen de ninguna solución tecnológica que permita un panorama descriptivo de las operaciones. Solo cuentan con un sistema de registro para la facturación de las operaciones desarrollado en Visual FoxPro útil en su momento, actualmente se requiere capacidad de integración con otros sistemas actuales para versatilidad, escalabilidad y seguridad de la información.

El propósito de este trabajo de investigación exploratorio, emprende un análisis para predecir los patrones de arribo de los buques para el traspaso de hidrocarburos en el Terminal Portuario Marítimo, utilizando técnicas de Machine Learning. Inicialmente entrega un panorama descriptivo con datos del puerto, luego desarrolla un modelo predictivo para remolcadores.

1.3 Solución propuesta

El Terminal Petrolero La Libertad en los servicios de operaciones portuarias requiere de herramientas tecnológicas para administrar y gestionar las actividades del puerto.

Para facilitar servicios portuarios y prevención de la contaminación petrolera, es necesario planificar la logística necesaria para la demanda, considerando el número limitado de dársenas y remolcadores. Machine Learning con un alcance de análisis descriptivo, proporcionaría un prototipo con modelos explicativos sobre las operaciones, tiempo de maniobras, etc. Esta solución se traza con la metodología KDD (Knowledge Discovery in databases) conocida ampliamente en proyectos de Ciencia de Datos, se detalla más adelante en el epígrafe Metodología.

Según los *papers* en *journals*, en el sector marítimo petrolero carecen soluciones ML que aborden la logística para satisfacer la demanda de servicios por medio una predicción. Con lo expuesto, esta investigación propone la siguiente interrogante ¿Los modelos de ML pueden anticiparme los remolcadores necesarios para atracar buques?

1.4 Objetivos

La temática de la propuesta de investigación se define como un Proyecto Conjunto Empresa-Academia, con un enfoque de investigación exploratorio y descriptivo. Corresponde a la línea de investigación es la Eficiencia energética, Energías renovables y Alternativas —de ODS 7 energía asequible y no contaminante—, perteneciente al área de Logística y Transporte.

A continuación, se detalla los objetivos a través de la taxonomía Krathwohl (2000) basando por el autor Bloom:

1.4.1 Objetivo General

Realizar un análisis descriptivo y predictivo para el abastecimiento de remolcadores en maniobras de traspaso de hidrocarburos en un Terminal Portuario Marítimo, utilizando técnicas de Machine Learning.

1.4.2 Objetivos Específicos

- (1). Identificar las variables endógenas y exógenas

- (2). Comprender e interpretar patrones y características en función del análisis descriptivo
- (3). Aplicar técnicas de Machine Learning para un análisis predictivo de los remolcadores
- (4). Determinar el modelado optimo a través de métricas de evaluación

1.5 Metodología

KDD (Knowledge Discovery in Databases) es un proceso sistemático cuyo objetivo es extraer conocimiento útil y comprensible a partir de grandes volúmenes de datos, mediante etapas como la selección, transformación, minería de datos y evaluación de resultados (Figura 2). La selección de la metodología KDD para este proyecto de *forecasting* se justifica por su capacidad para abordar de manera estructurada y rigurosa el descubrimiento del conocimiento. El cual contempla las etapas necesarias para obtener predicciones precisas y relevantes, tales como selección, limpieza, análisis, transformación, modelado, evaluación e interpretación de resultados. De esta manera, se asegura no solo la calidad del proceso analítico, sino también la utilidad práctica del conocimiento generado para la toma de decisiones futuras (Fayyad *et al.*, 1996).

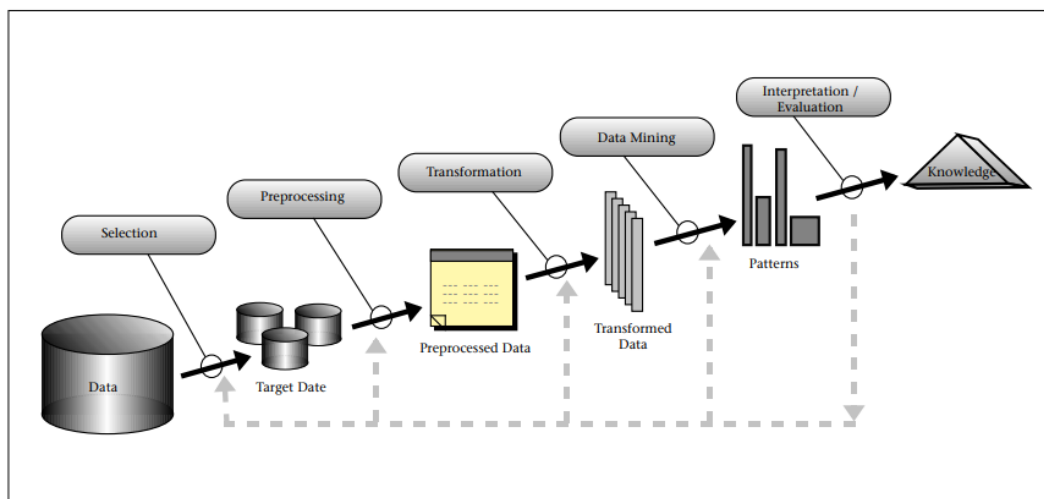


Ilustración 2. Componentes del proceso KDD. **Fuente:** Fayyad *et al.*, 1996

1.6 Dataset

El conjunto de datos se obtuvo de la Superintendencia del Terminal Petrolero La Libertad (SUINLI) para el propósito de investigación por medio de un acuerdo de confidencialidad y anonimato de los datos.

En las Tablas 1, 2 y 3 se dispone de información relacionada con los buques y remolcadores, la cual proviene del sistema de facturación asociado a las operaciones logísticas.

Tabla 1. Tabla arribo de buques con registros desde 1999

ABREVIATURA	DESCRIPCIÓN
REGISTRO *	Número de registro de la asistencia para la operación portuaria
MATRICULA	Número de identificación del Buque
TIPONAVE	Tipo de buque
NOMBRENAVE	Nombre del buque
BANDERA	País de procedencia
CHARTEADO	Buques internacionales que se encuentran fletados, Charteados por un período para operativos de EP PETROECUADOR.
FECHAARRIBO	Fecha cuando inicia la asistencia
HORAARRIBO	Hora cuando inicia la asistencia
FECHZARPE	Fecha cuando finaliza la asistencia
HORAZARPE	Hora cuando finaliza la asistencia
TRAFARRIBO	Tipo de buque al que se asiste: I para internacional. C para nacional llamado cabotaje.
TRAFZARPE	Tipo de buque al que se asiste: I para internacional. C para nacional llamado cabotaje.
AGENCIA	La agencia de la compañía naviera
PTOARRIBO	Terminal Portuario de Arribo
PROCEDENCIA	País de procedencia del buque
DESTINO	País donde se dirige el buque
CARGA	De la carga del buque el país de procedencia
FACTURA	Número de factura
BOC	Expresa cuando la asistencia solicitada es de carácter carga de combustible
IMPORTACION	Producto Internacional con la que arriba el buque para ser descargado en el puerto.
TOTALHORAS	Total horas de arribo hasta salida del buque
ESLORA	Largo del buque
TRBNAVE	Tonelaje de Registro Bruto. Peso del volumen de carga
TRNNAVE	Tonelaje de Registro Neto. Peso de la nave sin carga
TPMNAVE	Tonelaje de Peso Muerto. Es el total del peso del buque más el peso de la carga
RANGOTPM	Rango de tonelaje
CALADONAVE	Distancia vertical entre un punto de línea de flotación y la línea base o quilla, incluido el espesor del barco
CALADOARRIBO	Nivel de calado con que llega al puerto
CALADOZARPE	Nivel de calado con que sale del puerto

Nota: * columna Registro. Es el enlace entre todos los servicios, desde la llegada hasta la partida del buque en el Terminal.

Tabla 2. Tabla remolcadores con registros desde 2016

ABREVIATURA	DESCRIPCIÓN
COMPANIA	Compañía del remolcador
REMOLCADOR	Nombre del remolcador
RANGOHP	Potencia del remolcador
BP	Capacidad de empuje
RANGOBP	Rango de capacidad de empuje
CLASE	A: pequeño; B: mediano; C: grande
REGISTRO *	Número de registro de la asistencia para la operación portuaria
BUQUE	Tipo de buque al que se asiste: I para internacional. C para nacional llamado cabotaje.
MATRICULABUQUE	Matrícula del buque asistido
NOMBREBRUQUE	Nombre del buque asistido
TRBBUQUE	Tonelaje de Registro Bruto. Es el volumen total almacenado en el buque transportándose
AGENCIABUQUE	Agencia del buque
PRACTICAJE	Tipo de asistencia del practicaaje en la operación portuaria
BUQUE2	Nombre del buque asistido 2
LUGAR	Lugar de fondeo (estacionamiento) en el Terminal
FECHAINI1	Fecha cuando inicia la asistencia
HORAINI1	Hora cuando inicia la asistencia
FECHAFIN2	Fecha cuando finaliza la asistencia
HORAFIN2	Hora cuando finaliza la asistencia
TOTALHORAS	Hora total de asistencia en buques

TARIFABUQUE	Tipo de buque al que se asiste: I para internacional. C para nacional llamado cabotaje.
VAL. MONEDA	Valor de la factura
ANULADA	Cuando se anula la papeleta por error en ingreso (digitación de numeración o duplicado)

Nota: * columna Registro. Es el enlace entre todos los servicios, desde la llegada hasta la partida del buque en el Terminal.

Tabla 3. Características de los remolcadores

CLASE	RANGO H.P.	OPERADORA	BOLLARD PULL
A	<= 1500	OPERADOR1 S.A.	11
			15
		OPERADOR2 S.A.	11
			8
		OPERADOR3 S.A.	20
B	1501 a 3000	OPERADOR4 S.A.	10
		OPERADOR5 S.A.	18
		OPERADOR6 S.A.	29
		OPERADOR7 S.A.	29
		OPERADOR8 S.A.	31
		OPERADOR9 S.A.	29
		OPERADOR10 S.A.	25
C	>= 3000		50
		OPERADOR11 S.A.	60
			67
		OPERADOR12 S.A.	55
		OPERADOR13 S.A.	45

Nota: ** Remolcadores que no están operativos

Para el estudio de variables meteorológicas obtenidas desde estaciones terrestres, inicialmente se consideró el uso de la base de datos del Instituto Nacional de Meteorología e Hidrología (INAMHI) a través del Sistema de Estandarización de Datos Hidroclimáticos Crudos (SEDC). No obstante, dado que esta plataforma está orientada a estaciones ubicadas en la provincia de Pichincha, se optó por utilizar datos hidroclimáticos provenientes de la librería meteostat de Python, que proporciona registros históricos de estaciones meteorológicas a nivel global, incluyendo aquellas ubicadas en la zona de estudio.

Meteostat es un proveedor de datos meteorológicos y climáticos de acceso abierto que ofrece series temporales a largo plazo provenientes de miles de estaciones meteorológicas distribuidas a nivel mundial. Su enfoque en datos de observación reales —en lugar de simulaciones o modelos suprarregionales— lo convierte en una fuente confiable para estudios científicos, proyectos educativos y desarrollos tecnológicos. La plataforma se destaca por su política de datos abiertos, su amplia cobertura geográfica, y su facilidad de integración a través de una biblioteca específica para Python. Además, obtiene sus datos de diversas fuentes oficiales como la NOAA (National Oceanic and Atmospheric Administration) y el Deutscher Wetterdienst (DWD), asegurando calidad y consistencia conforme a las normas establecidas por la Organización Meteorológica Mundial (OMM). Estas características hacen de Meteostat una herramienta adecuada

para el análisis hidroclimático en contextos donde otras fuentes, como SEDC, presentan limitaciones de cobertura.

CAPÍTULO 2

2. ESTADO DE ARTE

2.1 Inteligencia Artificial (IA) en la logística marítima

Según la Asociación para el Avance de la Inteligencia Artificial (AAAI), la IA se define como “avanzar en la comprensión científica de los mecanismos subyacentes al pensamiento y el comportamiento inteligente y su encarnación en las máquinas”. Machine Learning (ML) es un subcampo dentro de la IA, a través del cual un sistema puede aprender y mejorar automáticamente a partir de datos sin estar programado explícitamente (Bhavsar citado por Filom *et al.*, 2022). El aprendizaje automático extrae conocimiento de los datos y realiza predicciones y decisiones informadas basadas en lo aprendido, sin necesidad de un conocimiento previo de los datos y el contexto (Hwarng y Ang citado por Filom *et al.*, 2022).

Una ventaja predominante de los modelos basados en ML es que, en muchos casos, el proceso de previsión podría automatizarse; por otro lado, los modelos estadísticos, como la Regresión Lineal (LR) y ARIMA, deben volver a realizarse periódicamente ya que los datos de entrada cambian constantemente (Önsel Ekici citado por Filom *et al.*, 2022). Además, a diferencia de la minería de datos o sistemas expertos, que se basaban en reglas predeterminadas para analizar los datos, el ML no se basa en reglas o ecuaciones actuales como modelo (Filom *et al.*, 2022).

El aprendizaje automático es una herramienta poderosa para el análisis predictivo y prescriptivo, mientras que métodos Investigación Operativa son más adecuados para el análisis prescriptivo (Crainic *et al.*, 2009). Por lo que, los métodos de ML están sustituyendo gradualmente estos enfoques para investigar problemas relevantes portuarios, tales como, predicción de la demanda, operaciones terrestres, operaciones costeras, seguridad, entre otras; dilatando nuevas vías para futuras investigaciones sobre puertos inteligentes (Filom *et al.*, 2022). En la Figura 3 muestra una vista panorámica de la analítica de negocios en las operaciones portuarias.

En los estudios recientes se destaca el cambio hacia la explotación de modelos híbridos, capaces de simular características tanto lineales como no lineales de los datos de entrada. Este tercer enfoque de modelo híbrido se trata de combinar modelos estadísticos con modelos ML, aprovechando las ventajas de ambos modelos para un mejor rendimiento de predicción (Ping y Fei citado por Filom *et al.*, 2022).

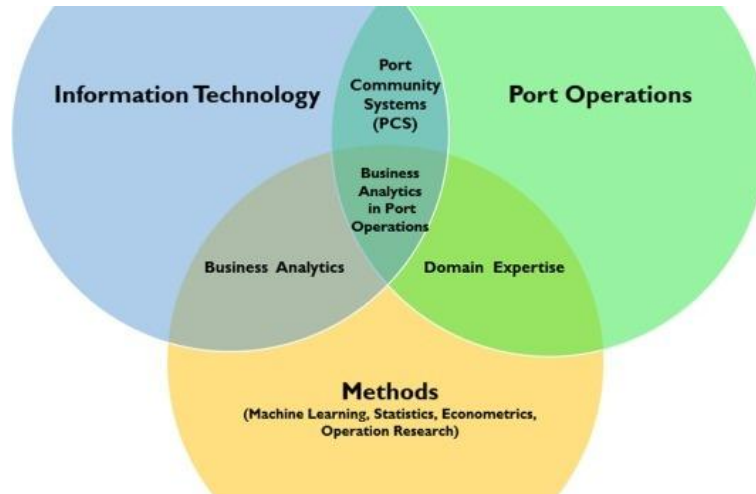


Ilustración 3. Vista panorámica de la analítica de negocios en las operaciones portuarias.

Fuente: Filom *et al.*, 2022

Aunque la implementación de métodos de aprendizaje automático en el transporte marítimo y la industria portuaria ha ido atrayendo gradualmente a académicos, solo unos pocos estudios ofrecieron revisar, categorizar y encontrar direcciones de investigación en la literatura relevante. Según Filom (2022) en su investigación sistemática sobre el aprendizaje automático en operaciones portuarias, con un análisis final de 70 *papers*, manifiesta que el transporte marítimo a granel supera en más del 50% al transporte contenedores, además de no encontrar un estudio sobre el pronóstico de la demanda de carga a granel utilizando métodos ML; tomando en cuenta que el transporte marítimo está intrínsecamente impulsado por la demanda, que afecta al resultado por factores socioeconómicos.

2.2 Trabajos de investigación relacionados

La bibliografía sobre puertos inteligentes es muy amplia y variada, sin embargo, con respecto a terminales petroleros es escasa. Para efectos de este trabajo de investigación se citan un par de casos relevantes relacionado a buques con ML:

- Kim y Lee (2019) desarrollaron un modelo DNN (Deep Neural Network) para predecir el destino de los buques en el puerto de Yeosu, Corea del Sur, se utilizó como datos

el área de navegación, tonelaje del barco, profundidad del barco, eslora, tipo y ocupación del puerto extraídos de AIS y del Sistema de Información de Gestión portuaria (MIS) basados en 2 años, logrando una precisión del 85%, superior en un 10-15% al modelo de referencia.

- En algunos puertos los canales de acceso para la navegación son limitados convirtiéndose en los activos más importantes del puerto. Por ejemplo, el puerto de Tianjin, China, su canal Dagusha es uno de los cuellos de botella más predominantes del puerto, por lo que una estimación realista de la capacidad es indispensable para los administradores portuarios. En el 2020 Liu *et al.* (citado por Filom *et al.*, 2022) calcularon la capacidad de navegación del canal Dagusha usando datos AIS y el método de agrupación K-means, identificando la capacidad máxima del canal, el resultado ayudó en la planificación de la llegada y salida de los barcos

Con respecto a remolcadores se encontró dos trabajos de investigación del 2020, estas soluciones no utilizan ML:

- Chen *et al.*, (2020) por medio de algoritmos analiza las actividades de los remolcadores utilizando datos AIS para el puerto Tiajin, China. Presenta algunas características como el número diario de trabajos y distribución espacial de los remolcadores, además realiza un análisis temporal y espacial para investigar la asignación de remolcadores, el tiempo de servicio, la utilización de los remolcadores y las ubicaciones de las operaciones de atraque y desatraque.
- Wu y Miao (2020) por medio de algoritmos presenta un modelo estadístico programación robusto, abordando al problema de asignación continua de atraques con incertidumbre por la hora de llegada de los buques y el tiempo de operación y maniobras.

Por último, se encontró un trabajo de postgrado orientado a remolcadores con ML de Borzyszkowski (2022), quien propone un modelo para estimar el número de remolcadores implicados en un evento operativo de remolque. Usaron datos históricos de un año con datos AIS de tipo dinámico sobre el tráfico marítimo y meteorológicos del mar Báltico en dos puertos, Rauma y Vuosaari. Logró construir modelos específicos por puerto con *overall accuracy* del 87% y 91.5% por medio de algoritmos de clasificación. Estos resultados demostraron que es posible desarrollar una herramienta predictiva para las operaciones de remolcadores y por ende coadyuvar a la eficiencia operativa en los puertos a través de IA.

2.3 Series temporales

Las series temporales son secuencias de datos recopilados en intervalos de tiempo específicos, que se utilizan para analizar y predecir tendencias futuras. Esta área de análisis se centra en identificar patrones, tendencias, estacionalidades y otros comportamientos cíclicos en los datos a lo largo del tiempo (Filom *et al.*, 2022).

Un pronóstico se define como una estimación de un valor futuro para una variable, con la ayuda de técnicas para minimizar el margen de error. Además, es fundamental aplicar el buen juicio y la experiencia del experto que genera el pronóstico. El objetivo de forecasting (pronóstico) es predecir eventos futuros basándose en datos históricos, con el fin de mejorar la planificación, la toma de decisiones y la asignación de recursos (Jiahuan *et al.*, 2023).

El objetivo del análisis de series temporales suele ser doble: comprender o modelizar el mecanismo estocástico que da lugar a una serie observada y, predecir o pronosticar los valores futuros de una serie basándose en la historia de esa serie y, posiblemente, en otras series o factores relacionados. La lista de ámbitos en los que se estudian las series temporales es prácticamente interminable (Cryer y Chan, 2008).

Una serie de datos es una secuencia de valores que representan la evolución de una sola variable a lo largo del tiempo o de algún otro criterio. Por ejemplo (Figura 4), si registramos la temperatura diaria durante una semana, estamos trabajando con una sola serie. En cambio, cuando se recopilan y analizan varias variables simultáneamente, como temperatura, humedad y velocidad del viento, en los mismos días, hablamos de múltiples series.

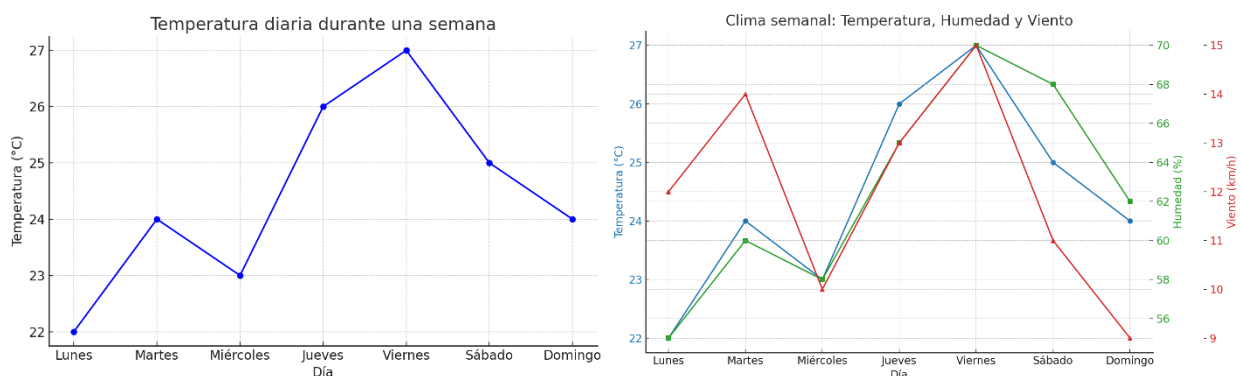


Ilustración 4. Ejemplo de uniSerie y multiSerie

2.3.1 Modelos de predicción en series temporales

El pronóstico o *forecasting* de variables cuantitativas puede abordarse mediante diversas técnicas de modelado predictivo, las cuales buscan representar el comportamiento histórico de una serie con el fin de anticipar sus valores futuros. Se agrupan en:

- **Modelos estadísticos y econométricos**, como ARIMA, SARIMA o regresiones dinámicas, los cuales parten de supuestos teóricos sobre el comportamiento de la serie y su estructura de dependencia temporal (Jiahuan *et al.*, 2023).
- **Modelos de *machine learning***, como árboles de decisión, bosques aleatorios y *gradient boosting*, que permiten capturar relaciones no lineales complejas sin necesidad de supuestos paramétricos estrictos.
- **Modelos de *deep learning***, como redes neuronales recurrentes (RNN), diseñados para aprender patrones secuenciales a gran escala en contextos donde existen múltiples fuentes de variabilidad y datos de alta dimensionalidad.

La elección de una u otra técnica depende del contexto del problema, la disponibilidad de datos, la interpretabilidad deseada y el grado de precisión requerido. En este sentido, Hyndman y Athanasopoulos (2018) enfatizan que: cuando la serie tiene patrones fuertes (varianza explicada por tendencia/estacionalidad), los modelos simples (estadísticos y econométricos) funcionan bien; cuando hay alta incertidumbre (ruido) o factores externos no observados, están los modelos con regresores (ML) o técnicas más complejas (DL).

2.3.2 Forecasting

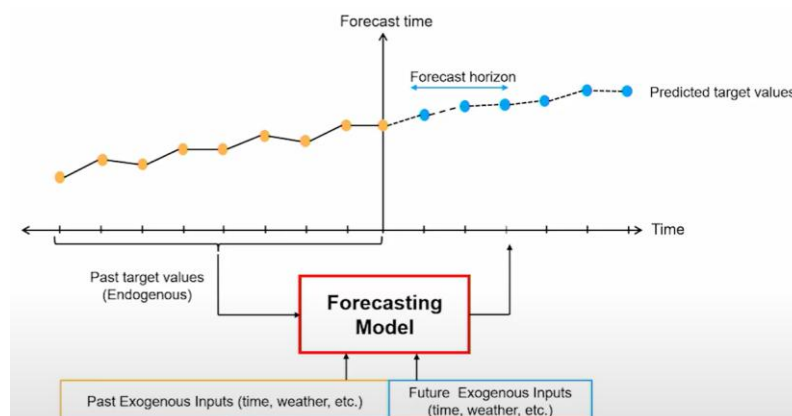


Ilustración 5. Contexto sobre Forecasting Model. **Fuente:** [Abhishek Murthy](#)

Consiste en construir una representación matemática basada en datos históricos, con el objetivo de predecir los valores futuros de una serie temporal. En un modelo de

forecasting, los valores pasados (endógenos) de la variable objetivo se utilizan para identificar patrones históricos, mientras que los *inputs* exógenos pasados (como clima, tiempo o eventos) aportan contexto adicional al comportamiento de la serie. Los inputs exógenos futuros, conocidos de antemano (e.g. feriados, fines de semana), enriquecen las predicciones al incorporar información predecible anticipada. A partir del tiempo de pronóstico (*forecast time*), el modelo proyecta los valores hacia adelante según un horizonte de predicción (uno o varios pasos) definido, generando los valores pronosticados. Todo este proceso es gestionado por el modelo de *forecasting*, que integra estas entradas para estimar el comportamiento futuro de la variable de interés (Figura 5).

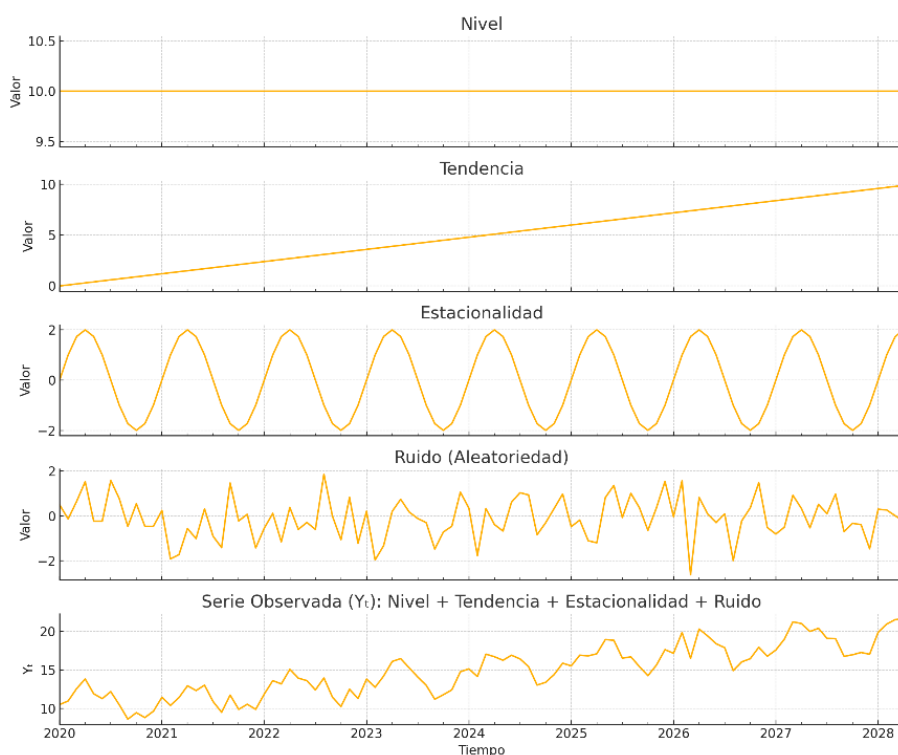


Ilustración 6. Componentes de una serie temporal

Este modelado puede realizarse considerando únicamente los patrones internos de la propia serie (modelo autorregresivo) o incorporando variables externas que influyan en el comportamiento de la variable de interés. De forma simplificada se puede representar:

$$Y_t = \text{Nivel} + \text{Tendencia} + \text{Estacionalidad} + \text{Ruido}$$

Cada uno de estos elementos cumple una función específica (Figura 6):

- Nivel (*level*): representa el valor base alrededor del cual fluctúa la serie: ejemplo, el promedio mensual de temperatura sin considerar la variación estacional.

- Tendencia (*trend*): refleja la dirección general del cambio a largo plazo (creciente o decreciente).
- Estacionalidad (*seasonality*): corresponde a patrones que se repiten en intervalos regulares (como días, semanas o años).
- Ruido (*noise*): incluye toda variación aleatoria no explicada por los otros componentes. Corresponde la parte impredecible de la serie.

Desde una perspectiva orientada al modelado predictivo en *machine learning*, esta descomposición puede reinterpretarse como:

$$\text{FORECAST} = \text{Patrones} + \text{Varianza no explicada}$$

Patrones (Endógenas): Abarca todos los elementos repetitivos o sistemáticos que pueden ser aprendidos por el modelo. Para capturar estos elementos estacionales, tendencias y nivel, a continuación, algunas de las técnicas más usadas:

- Retrasos Temporales (*lags*): Son valores pasados de la misma serie que permiten capturar la memoria, es decir, cómo el valor actual depende de valores pasados.
- Estadísticas móviles (*rolling stats*): Son características estadísticas calculadas por medio de una ventana deslizante de valores pasados en una serie temporal. Estas ventanas capturan comportamientos locales, como la media, desviación estándar, mínimo o máximo recientes, proporcionando información sobre los patrones temporales sin perder la secuencia cronológica. Por ejemplo, una media móvil de 7 días ayuda a suavizar la serie y resaltar tendencias de corto plazo.
- Diferencias y transformaciones internas: Ayudan a estabilizar tendencia y varianza en la serie, revelando estructuras como crecimiento o estacionalidad oculta.

Varianza no explicada (Exógenas): Representa el componente de incertidumbre asociado a factores que afectan la variable respuesta, pero no están reflejados directamente en el pasado de la serie, ayudan a reducir la varianza no explicada, siendo:

- Variables calendario-administrativas: Día de la semana, feriados, turnos laborales. No forman parte de la serie temporal directa.
- Eventos meteorológicos, económicos, sociales: Son externos y no se reflejan en los lags ni rolling stats. Ej.: tormentas, huelgas, ferias, anuncios de política
- Aleatoriedad estructural / ruido blanco / errores: Parte impredecible del sistema. La varianza no explicada incluye ruido blanco (azar puro), aleatoriedad estructural

(influencias no observadas o no modeladas) y los errores residuales del modelo.

Define los límites del poder predictivo y justifica el uso de intervalos de predicción.

La probabilidad de éxito en el pronóstico mejora cuando los patrones explican una mayor proporción de la variabilidad observada. En cambio, a medida que la varianza no explicada cobra mayor relevancia, se hace necesario incorporar variables exógenas — si están disponibles— para mejorar la capacidad predictiva del modelo y capturar mejor el comportamiento histórico observado.

2.4 Modelos *Machine Learning* para *Forecasting*

2.4.1 Introducción a los árboles de decisión

Los árboles de decisión son algoritmos predictivos en el que está basado en decisiones binarias (si/no) en función de varios atributos -que interactúan de forma no lineal- de una observación/variable respuesta. Estos no solo se destacan por su precisión en las tareas de clasificación -predice una etiqueta de clase- (Figura 7) y regresión -predice un valor numérico continuo- (Figura 8), los árboles de decisión son interpretables, requieren poca preparación de datos, funcionan bien en contextos donde se necesita trazabilidad o explicabilidad, y sobre todo, posee algoritmos *ensemble* que combinan muchos árboles simples en una colección de árboles combinados (*ensemble*) convirtiéndose en un predictor preciso, robusto y generalizable.

Una de las desventajas en los árboles de decisión, es su incapacidad de extrapolar fuera del rango de predictores observados durante el entrenamiento, lo cual inicialmente es una limitación para forecasting. Esto quiere decir, si el modelo entrenado con variables independientes entre 45 y 100, el árbol no sabrá cómo comportarse al querer una predicción para 20 o 140; en este caso, el árbol solo puede asignar predicciones basadas en divisiones ya aprendidas, por lo que repite valores conocidos (constantes) en lugar de proyectar una tendencia creciente o decreciente. En la Figura 9 se aprecia la diferencia entre modelos de árboles de regresión y modelos lineales en la extrapolación.

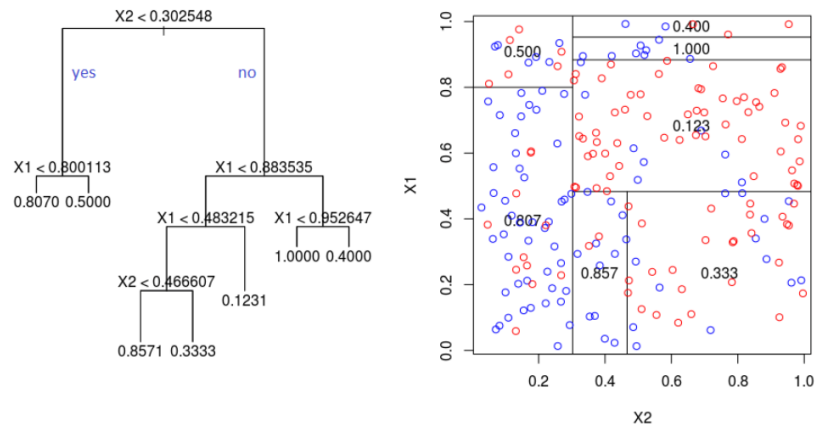


Ilustración 7. Árboles de Decisión Clasificación. Fuente: https://fhernanb.github.io/libro_mod_pred/arb-de-regre.html

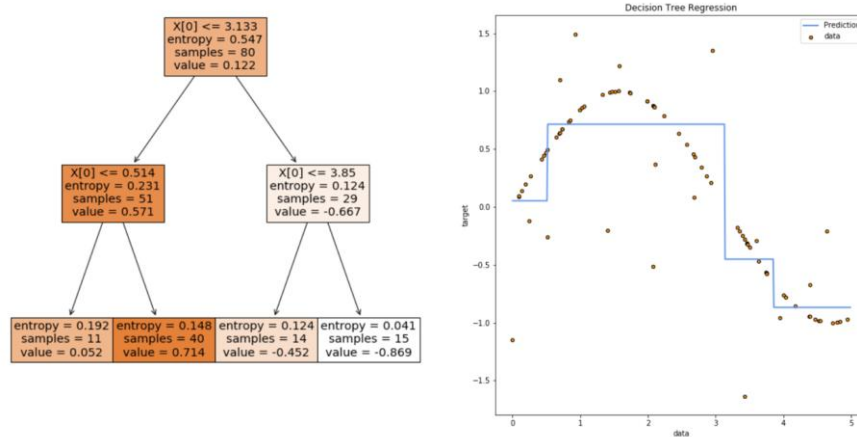


Ilustración 8. Árboles de Decisión Regresión. Fuente: <https://www.cristobal-aguirre.com/arboles-de-decision>

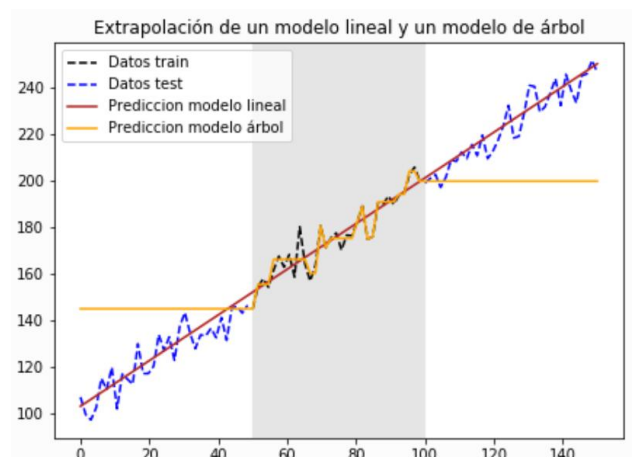


Ilustración 9. Extrapolación con modelos de árboles y modelos lineal. Fuente: https://cienciadedatos.net/documentos/py07_arboles_decision_python

2.4.2 Árboles de decisión: Extrapolación y Forecasting

La extrapolación en los árboles de decisión tradicionales es ingenua cuando se trata de valores futuros o fuera de rango del conjunto de entrenamiento, el árbol tiende a repetir (valor constante) la salida con el último valor sin proyectar tendencias o estacionalidad, lo cual restringe su utilidad directa en tareas de pronóstico (Figura 9). No obstante, en la actualidad se ha desarrollado nuevas estrategias en el ecosistema de Python para adaptar los árboles de regresión a problemas de *forecasting* con escenarios de pronóstico complejos y realistas. La extrapolación en modelos predictivos se refiere a la capacidad de anticipar valores fuera del rango observado en los datos históricos. Este proceso se fundamenta en la suposición de que los patrones pasados son representativos del comportamiento futuro del sistema. En series temporales (forecasting), la extrapolación es crítica, porque constantemente se predicen valores futuros no observados (Tabla 4).

Tabla 4. Diferencias entre árboles tradicionales y árboles forecasting

	ML en datos tabulares (regresión y clasificación)	ML en datos tabulares (pronóstico/forecasting)
División de entrenamiento/prueba	Asignación aleatoria.	División según el tiempo.
Creación de características y objetivo	Se pueden precomputar las características y el objetivo antes del momento de predicción.	Las características se construyen a partir del objetivo generado "bajo demanda" en el momento de predicción para el conjunto de prueba.
Predicción	Solo se necesita el modelo entrenado en el momento de la predicción.	Se necesita el modelo entrenado y el conjunto de entrenamiento en el momento de la predicción.
Ingeniería de características	No aplica	Es específica para series temporales e implicaciones de fugas de datos

2.4.3 Algoritmos combinados (*ensemble*)

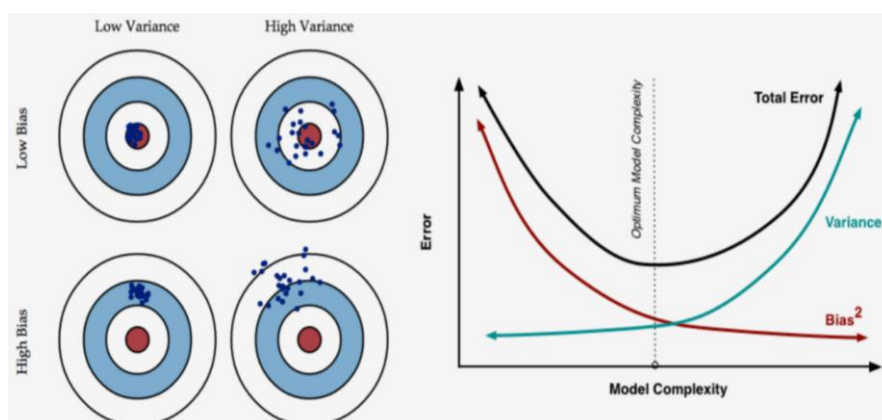
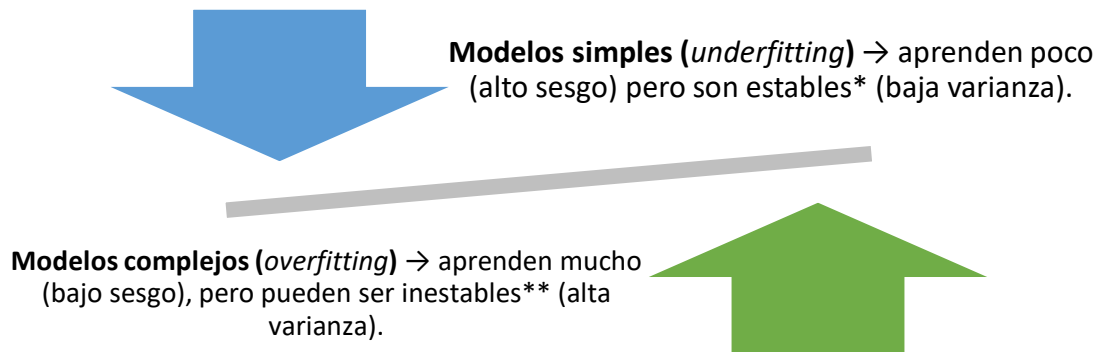


Ilustración 10. Error total de un modelo en *bias + variance + ϵ* . Fuente: [Link](#)

El sesgo (bias) indica cuánto se alejan, en promedio, las predicciones de un modelo respecto a los valores reales; si el modelo es demasiado simple, no capta bien la relación entre variables y su sesgo es alto. La varianza, en cambio, muestra cuánto cambian las

predicciones del modelo si se utiliza diferentes datos de entrenamiento (Figura 10). Los modelos basados en árboles pueden sufrir de ambos problemas *underfitting* como el *overfitting*, según el tamaño. Para equilibrarlos, se usan métodos *ensemble* como *bagging* y *boosting*, que combinan varios modelos para mejorar la precisión y estabilidad.



NOTA: *Estable se refiere a que un modelo simple, no cambia mucho si se entrena con diferentes datos, por lo que tiene baja varianza.

**Inestables se refiere a la flexibilidad para ajustarse, incluso a patrones sutiles o ruido, esto lo hace sensible a cambios en los datos: si se entrena con otro subconjunto, podría dar predicciones muy diferentes. Esto significa alta varianza.

Aunque tanto *bagging* como *boosting* buscan mejorar la precisión del modelo, lo hacen de forma diferente: a) **bagging** reduce el error al disminuir la **varianza**, combinando muchos modelos entrenados con datos distintos. b) **boosting** reduce el **sesgo** (bias), corrigiendo los errores de un modelo tras otro de forma secuencial.

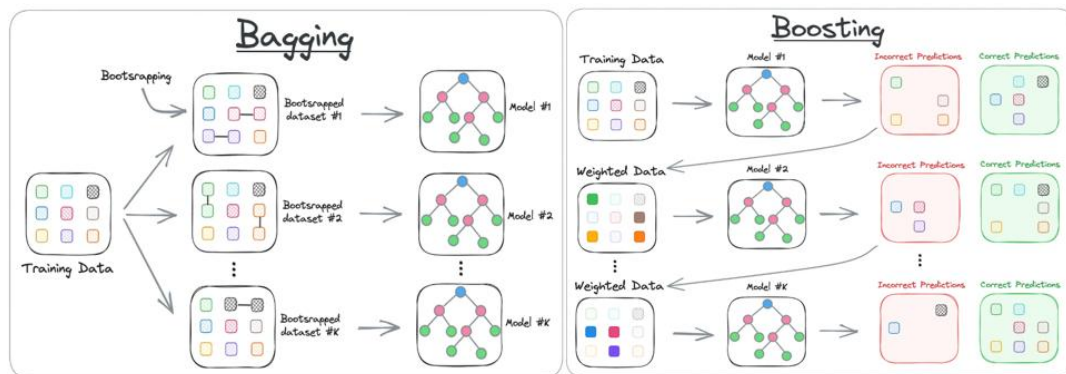


Ilustración 11. Algoritmos *ensemble*: Bagging & Boosting

En *bagging* los modelos se entrenan en paralelo con muestras aleatorias, mientras que en *boosting* cada modelo se ajusta con base en los errores del anterior (Figura 11). Las principales técnicas de *boosting* incluyen AdaBoost, que ajusta pesos para corregir errores; *Gradient Boosting*, que minimiza el error mediante gradientes; y variantes modernas como XGBoost, LightGBM y CatBoost, que optimizan el rendimiento con mejoras en velocidad, manejo de datos categóricos y regularización. Todas comparten el enfoque secuencial para combinar modelos débiles y mejorar la precisión.

2.4.3.1 Gradient Boosting

Boosting es una técnica general de ensamblado secuencial, mientras que *gradient boosting* es una variante específica que optimiza los errores usando descenso por gradiente. Un modelo *gradient boosting* está compuesto por varios de árboles de decisión (*ensemble*), el proceso de entrenamiento es secuencial, cada nuevo árbol aprende de sus predecesores obteniendo un menor error (residuo) en cada iteración (Figura 12).

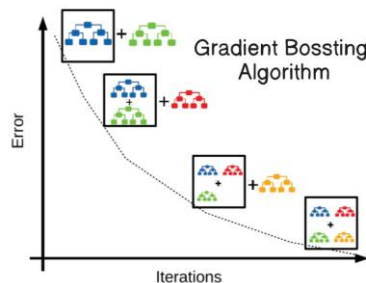


Ilustración 12. Proceso de los algoritmos *Gradient Boosting*. Fuente: [Link](#)

Este algoritmo tiene varias ventajas: seleccionan automáticamente los predictores más relevantes, funcionan tanto para regresión como clasificación, admiten variables numéricas y categóricas sin codificación previa (según la implementación), y al ser no paramétricos, no requieren supuestos de distribución. Además, necesitan poco preprocesamiento, son resistentes a valores atípicos, pueden predecir incluso con datos faltantes, ayudan a identificar variables importantes y escalan bien con grandes volúmenes de datos. Entre sus desventajas, los modelos de árboles múltiples pierden interpretabilidad frente a un solo árbol y no pueden extrapolar más allá del rango de los datos de entrenamiento.

2.5 Técnicas de *Gradient Boosting* adaptadas a *forecasting*

Los modelos de *gradient boosting* son populares en *machine learning* por el buen rendimiento en tareas de clasificación y regresión, consolidándose como una de las principales opciones para trabajar con datos tabulares, lo que ha dado lugar a diversas implementaciones. Una de las implementaciones son los árboles de regresión orientados a *forecasting* donde se adaptan algunos algoritmos de *gradient boosting*, en que destaca su habilidad para modelar relaciones no lineales entre variables, su alta escalabilidad para manejar grandes volúmenes de datos. Esta investigación aborda solo dos de ellas:

2.5.1 XGBoost (Extreme Gradient Boosting)

Es una implementación optimizada de gradient boosting que utiliza árboles de decisión como base. Se destaca por su eficiencia computacional y su capacidad para manejar datos dispersos y grandes volúmenes de información. Incorpora técnicas avanzadas como regularización L1/L2, poda inteligente de árboles y manejo de valores perdidos. La fortaleza principal es el rendimiento predictivo con capacidad de controlar el sobreajuste gracias a su regularización integrada.

2.5.2 LightGBM (Light Gradient Boosting Machine)

Desarrollado por Microsoft, LightGBM utiliza una estrategia de crecimiento de árboles "leaf-wise" y una técnica basada en histogramas para acelerar el entrenamiento. Es altamente eficiente en términos de memoria y velocidad, especialmente en conjuntos de datos grandes y de alta dimensionalidad. También soporta de forma nativa variables categóricas. La fortaleza principal es su capacidad para manejar grandes volúmenes de datos con alta velocidad de entrenamiento lo hace ideal para aplicaciones en tiempo real y sistemas de recomendación.

2.6 Biblioteca SKForecast

Skforecast es una biblioteca que simplifica el pronóstico de series temporales utilizando modelos de *machine learning*, integrando de forma fluida regresores compatibles con scikit-learn. Una de las principales fortalezas es su capacidad para incorporar tanto variables autorregresivas como variables exógenas de forma sencilla, permitiendo así capturar información relevante del entorno. Sin embargo, presentan retos: la necesidad de reestructurar los datos como problema de regresión; usar procesos iterativos para predicciones futuras; y aplicar métodos especiales de validación (*backtesting*), ya que la validación tradicional -K-Fold- es inadecuada para series temporales.

2.6.1 Forecaster: objeto orquestador y estrategias de predicción

En skforecast, un *Forecaster* es el objeto orquestador (implementa y coordina) que encapsula todo el flujo de pronóstico con estimadores tipo *scikit-learn* (XGBoost, LightGBM). Su propósito es estandarizar y automatizar la preparación de datos (dataset supervisado), el entrenamiento y la inferencia (uno o varios pasos) en series temporales, permitiendo comparar rigurosamente estrategias recursivas y directas, con o sin exógenas y soporte en pronóstico puntual o probabilístico (Tabla 5). Rol del Forecaster:

- Ingeniería de variables de serie temporal: creación de *lags*, *window features* y manejo de variables exógenas.
- Transformaciones y consistencia: diferenciación/escala y posterior *backtransform* para volver a la unidad original.
- Entrenamiento y predicción: ajuste/actualización del estimador y generación del horizonte de pronóstico.
- Evaluación y utilidades: *backtesting*, búsqueda de hiperparámetros y, opcionalmente, predicción probabilística (intervalos por bootstrap/quantiles).

Tabla 5. Tipos de Forecaster y sus características para modelos *gradient boosting*

Forecaster	Serie única	Serie múltiples	Estrategia recursiva	Estrategia directa	Predicción probabilística	Diferenciación de series temporales	Características exógenas	Window features
ForecasterRecursive	X	-	X	-	X	-	X	X
ForecasterDirect	X	-	-	X	X	-	X	X
ForecasterRecursiveMultiSeries	-	X	X	-	X	X	X	X
ForecasterDirectMultiVariate	-	X	-	X	X	-	X	X

En pronóstico de series temporales, existen dos estrategias ampliamente utilizadas para abordar horizontes de predicción múltiples: la estrategia recursiva y la estrategia directa. La **estrategia recursiva** consiste en entrenar un modelo para predecir un único paso hacia adelante, el valor en $t+1$. Y en el caso, de querer predecir varios pasos futuros como $t+2$, $t+3$, ..., el modelo reutiliza sus propias predicciones anteriores como entradas (*lags*) para continuar proyectando hacia el futuro, y así sucesivamente (Figura 13). Esta metodología permite reutilizar el mismo modelo para todo el horizonte, lo que la hace computacionalmente eficiente; sin embargo, tiende a acumular errores, ya que cada nueva predicción depende de la anterior, afectando la precisión. Por otro lado, la **estrategia directa** aborda cada paso del horizonte de manera independiente. Es decir, se entrena un modelo diferente para predecir $t+1$, otro para $t+2$, y así sucesivamente (Figura 14). Esto evita la propagación de errores entre pasos, mejorando la precisión en horizontes largos. No obstante, implica costo computacional y mayor complejidad en el entrenamiento, ya que se ajusta y mantiene múltiples modelos.

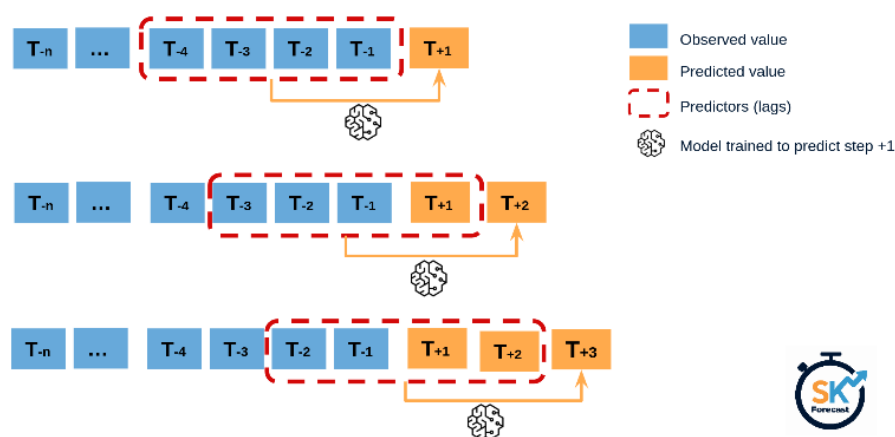


Ilustración 13. Estrategia Recursive. Fuente: <https://skforecast.org/0.16.0/introduction-forecasting/introduction-forecasting.html>

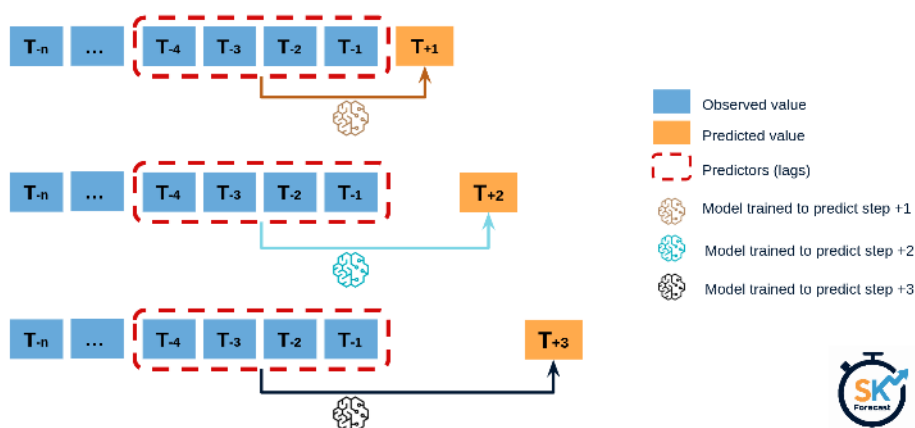


Ilustración 14. Estrategia Direct. Fuente: <https://skforecast.org/0.16.0/introduction-forecasting/introduction-forecasting.html>

La elección entre ambas estrategias dependerá del equilibrio deseado entre precisión, carga computacional y complejidad del problema (Tabla 6).

Tabla 6. Forecaster: Recursive Vs. Direct

Características	Estrategia Recursive	Estrategia Direct
Número de modelos	1 por serie	1 por horizonte
Usa predicciones anteriores	Si	No
Coste computacional	Bajo	Alto
Acumulación de error	Si	No
Precisión por paso	Menor a largo plazo	Alta por paso
Cada predicción	Depende del anterior	Es independiente

2.6.2 Transformación de la serie temporal para *forecasting*

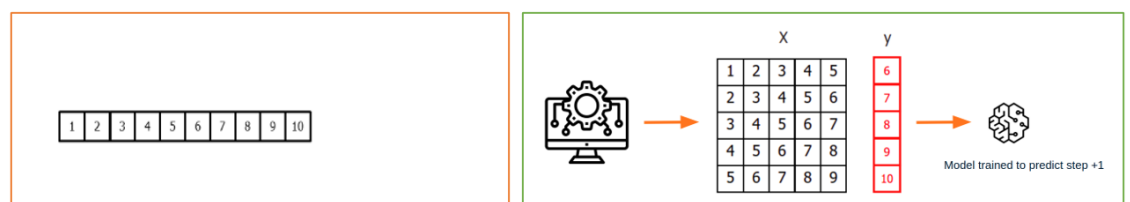
En el análisis de series temporales o datos secuenciales, es común que los datos no estén estructurados de forma supervisada, es decir, sin una variable objetivo explícita definida. Para aplicar modelos de aprendizaje supervisado —como redes neuronales

recurrentes o, árboles de regresión— es necesario transformar estos datos en un formato supervisado, donde se establezca una relación entre entradas (*features*) y salidas (*targets*). Este proceso consiste en **reformular el problema** de manera que el valor futuro de la serie (*target*) se prediga a partir de valores pasados (*features*).

En esta investigación centrada árboles de decisión, la transformación es un paso esencial en **forecasting moderno**, ya que permite aprovechar la potencia de los algoritmos supervisados en contextos originalmente no supervisados. A continuación, dos puntos importante:

1. Conversión a un *dataset* supervisado
2. Emplear técnicas Ingeniería de Características (*time series feature engineering*)

La conversión a un *dataset* supervisado consiste en reorganizar los datos para que exista una relación definida entre las variables independientes (x) —representa las entradas o *features*— y, la variable dependiente (y) —representa el *target* o valor que se desea predecir— (Figura 15). Este formato supervisado permite aplicar algoritmos de aprendizaje supervisado, que aprenden patrones en los datos para anticipar el valor de 'y' a partir de 'x'. En cambio, para los valores pasados (*features*) se aplica técnicas de ingeniería de características, esto es generar variables de retraso (*lags*), diferencias, ventanas móviles u otras características derivadas del tiempo (Figura 16).



Dataset (serie de tiempo) univariado NO supervisado

Dataset univariado supervisado con horizonte de predicción de 1 paso

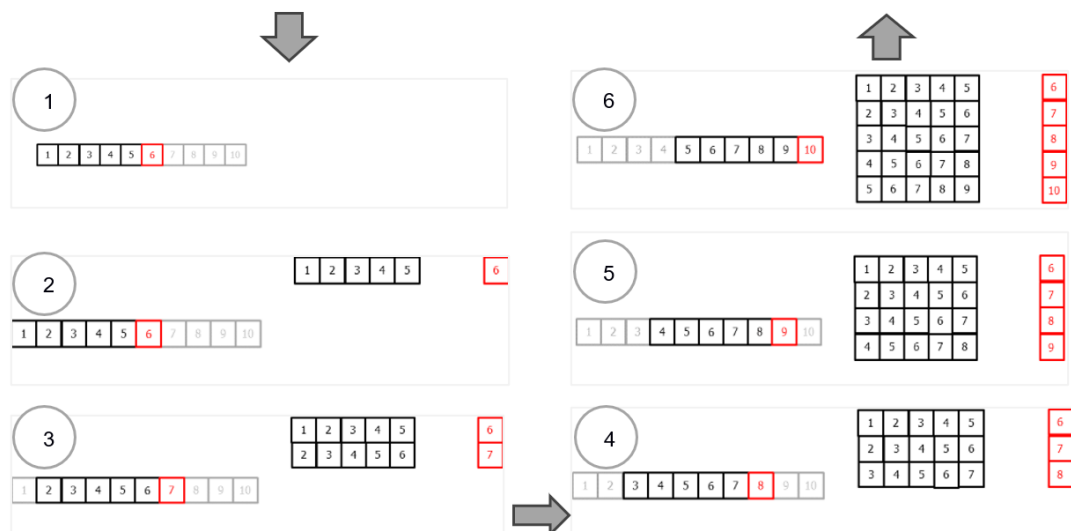


Ilustración 15. Conversión de un *dataset* NO supervisado a supervisado. Fuente: Adaptado de [Skforecast](#)

Time series

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

Exogenous variable

a	b	c	d	e	f	g	h	i	j
---	---	---	---	---	---	---	---	---	---

X

1	2	3	4	5	f
2	3	4	5	6	g
3	4	5	6	7	h
4	5	6	7	8	i
5	6	7	8	9	j

y

6
7
8
9
10

Ilustración 16. Transformación de series incluyendo una variable exógena. Fuente: [Skforecast](#)

Existen algunas técnicas de *feature engineering* para obtener los patrones y la varianza no explicada (exógenas) de la serie temporal. A continuación:

Patrones o Variables endógenas (varianza explicada)

- Lags (retrasos temporales): Capturan la memoria del sistema, es decir, cómo el valor actual depende de valores pasados (Figura 17). Ej.: $y[t-1]$, $y[t-2]$, ..., $y[t-n]$
- Rolling stats (ventanas móviles): Capturan comportamientos locales y suavizados, como tendencia a corto plazo (Figura 17). Ej.: $\text{mean}(y[t-7:t-1])$, std, min, max
- Transformaciones internas: Suaviza el crecimiento exponencial de la serie. Estabiliza la varianza, útil cuando los valores aumentan (Figura 18). Ej.: $\log(y)$
- Diferenciación: Convierte una serie no estacionaria en estacionaria. Elimina la tendencia convirtiendo en una serie centrada en cero (Figura 18). Ej.: $\text{diff}(y)$

Variables exógenas (varianza no explicada)

- Variables temporales: creadas a partir del tiempo para capturar patrones estacionales o comportamiento humano, como: días festivos, calendario, luz solar.
- Clima histórico: son variables que describen las condiciones meteorológicas pasadas, tales como: temperatura, precipitación, presión atmosférica.
- Codificación cíclica: en variables temporales como la hora, el día o el mes tienen una naturaleza cíclica que los modelos no captan por sí solos, por ejemplo, la hora 23 y la 0 son cercanas en el tiempo, pero no numéricamente. Para representar correctamente estos ciclos, se utiliza funciones seno y coseno, permitiendo al modelo reconocer patrones temporales repetitivos para mejorar su predicción.
- Polinomiales: consisten en crear nuevas variables combinando entre sí variables externas elevadas a potencias (cuadrados, productos cruzados, etc.) para capturar relaciones no lineales entre ellas. Por ejemplo, puede que la demanda de energía no aumente linealmente con la temperatura, pero sí lo haga con la interacción de temperatura y humedad.

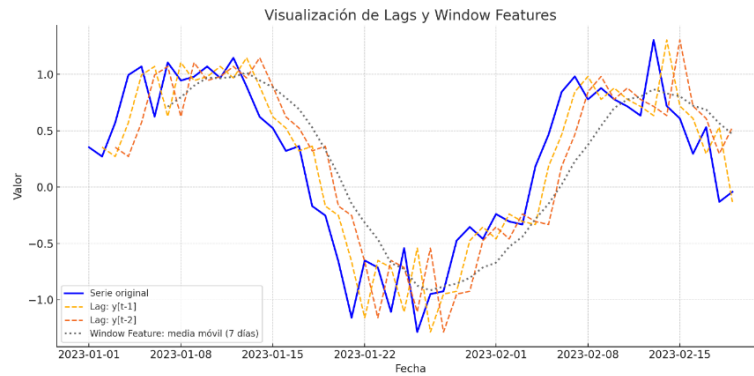


Ilustración 17. Visualización de los Lags y Window Features

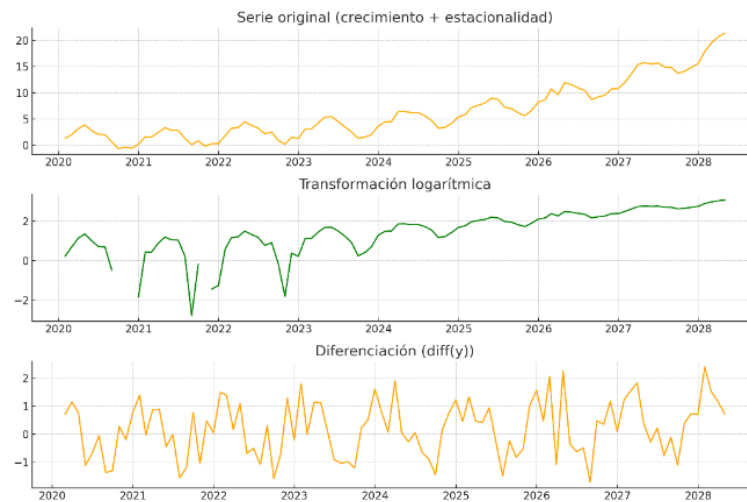


Ilustración 18. Visualización de las diferencias y transformaciones a una serie temporal

2.6.3 Backtesting

En aprendizaje automático la **validación cruzada (cross-validation)** es una técnica esencial para estimar el rendimiento de un modelo sobre datos no vistos. Su objetivo es reducir el riesgo de sobreajuste (*overfitting*) y garantizar que el modelo generalice más allá del conjunto de entrenamiento. Consiste en repetir el proceso de entrenamiento y evaluación del modelo utilizando diferentes particiones del conjunto de datos, de modo que todas las observaciones sean utilizadas tanto para entrenar como para validar el modelo en distintas rondas. Al generar múltiples combinaciones, se obtiene una estimación robusta y menos dependiente de una sola división de datos.

Una de las formas más comunes de validación cruzada es la **K-Fold Cross-Validation**, donde los datos se dividen en K bloques o "folds", y el modelo se entrena y evalúa K veces, usando un fold diferente como conjunto de prueba en cada iteración. Sin embargo, esta técnica asume que los datos son independientes entre sí, por lo que no es adecuada ya que rompe el orden cronológico. En estos casos, es necesario utilizar métodos que

respeten la estructura secuencial de los datos, como el **backtesting**, también conocido como **validación deslizante o walk-forward validation**, que permite simular cómo habría funcionado el modelo al predecir en distintos puntos del pasado (Figura 19).

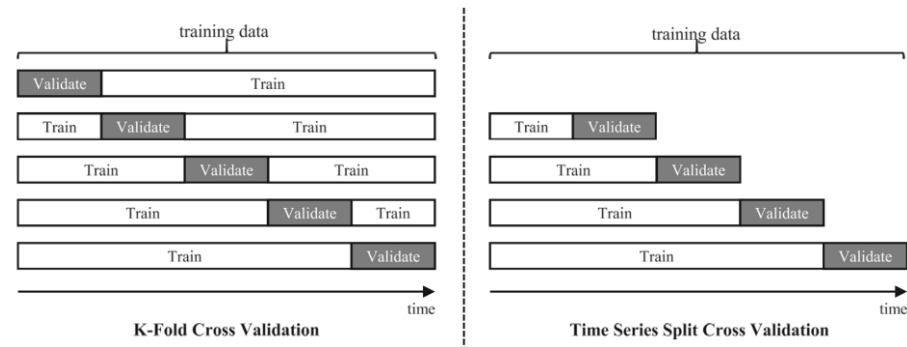


Ilustración 19. Cross-Validation: K-fold Vs. Backtesting. **Fuente:** [Rokoss et al., 2024](#)

Skforecast implementa este proceso de manera automatizada mediante la función `backtesting_forecaster()` para series univariadas y `backtesting_forecaster_multiseries()` para multivariables, la cual permite controlar parámetros como el tamaño del entrenamiento inicial `initial_train_size`, el número de pasos futuros a predecir `steps` llamado también horizonte de predicción (Figura 5) y, la estrategia de actualización del modelo `refit`, facilitando una evaluación realista y reproducible del desempeño en tareas de pronóstico (Tabla 7).

Tabla 7. Parámetros de Backtesting

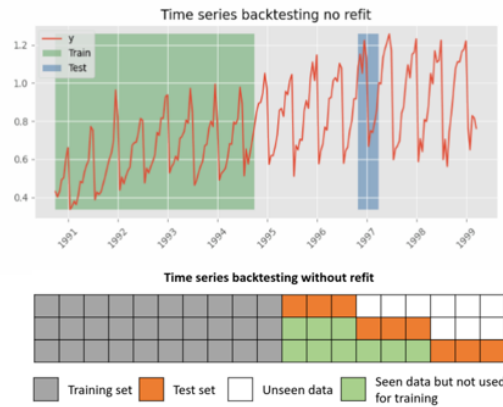
	<code>fixed_train_size = False</code>	<code>fixed_train_size = True</code>
<code>refit = False</code>	(1) Entrenado una sola vez con los datos iniciales (<code>initial_train_size</code>). Se usa para hacer predicciones sobre los datos disponibles. <code>'fixed_train_size'</code> no aplica.	
<code>refit = True</code>	(2) Ventana creciente (expanding window) que aumenta en cada iteración según el horizonte	(3) Ventana deslizante (sliding window) con paso igual al horizonte de predicción

Backtesting proporciona un enfoque estructurado para este tipo de evaluación, con dos parámetros: `'refit'` y `'fixed_train_size'`. El parámetro `'refit'` determina si el modelo se reentrena en cada iteración del proceso de validación. Si `fixed_train_size=False`, se emplea una ventana creciente (*expanding window*), donde el tamaño del conjunto de entrenamiento se amplía progresivamente. En cambio, si `fixed_train_size=True`, se aplica una ventana deslizante (*sliding window*), manteniendo el tamaño constante del conjunto de entrenamiento, el cual se desplaza en el tiempo. Esta configuración permite simular escenarios realistas en producción, ajustando la estrategia de validación según las características y requerimientos del modelo y del dominio de aplicación (Figura 20).

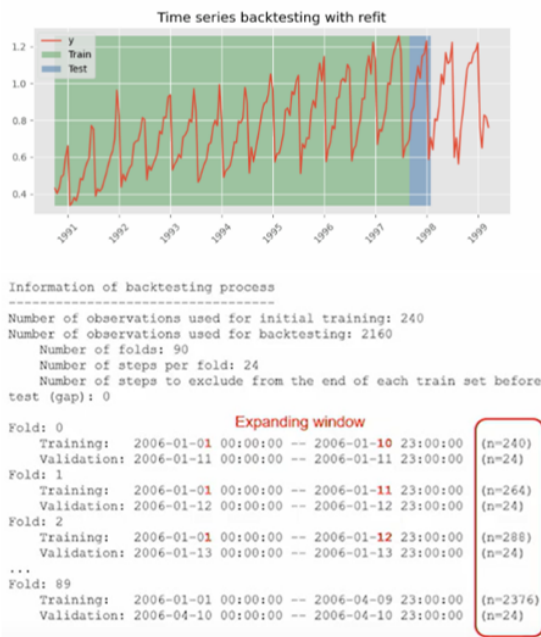
En la Sección 3.5, Hewamalage *et al.* (2023) destacan la importancia de evitar el *data leakage* durante la evaluación de modelos de pronóstico. Este fenómeno ocurre cuando, directa o indirectamente, se emplean valores futuros al calcular métricas de error, lo cual puede invalidar la evaluación al dar una impresión artificialmente optimista del rendimiento del modelo. Una recomendación clave de los autores es que las métricas deben calcularse secuencialmente, utilizando únicamente la información disponible hasta el momento de cada predicción.

La técnica de *backtesting* implementada por la función `backtesting_forecaster` o `backtesting_forecaster_multiseries` cumple con estos criterios. Permite evaluar modelos entrenados de forma única (`refit=False`) o reentrenados en cada paso (`refit=True`), siempre utilizando solo los datos disponibles hasta cada punto de predicción. Tanto el entrenamiento, la generación de predicciones y el cálculo de errores se realizan sin acceder a información futura, garantizando una evaluación temporalmente coherente y libre de filtraciones de datos. Este enfoque permite calcular métricas de rendimiento para cada bloque de tiempo, facilitando el análisis multi-horizonte. En resumen, el *backtesting* en *Skforecast* es una herramienta fundamental para validar la estabilidad y generalización de un modelo en escenarios temporales, respetando la lógica de la predicción futura basada únicamente en información pasada.

(1) Backtesting sin reentrenamiento



(2) Backtesting con reentrenamiento



(3) Backtesting con reentrenamiento y tamaño de entrenamiento constante

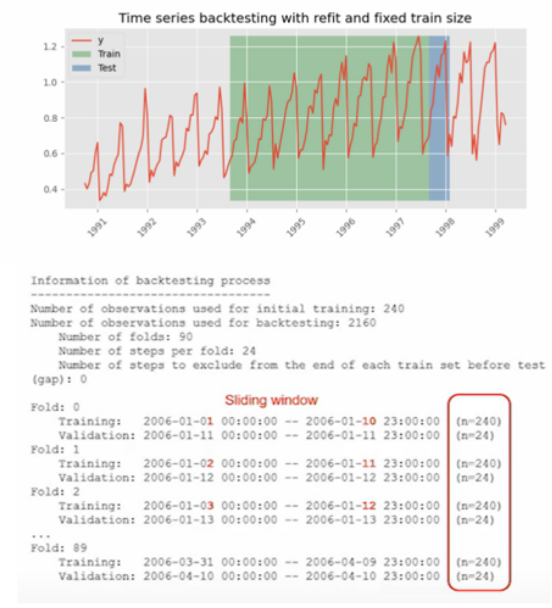


Ilustración 20. Ilustración de los parámetros de Backtesting de la Tabla 6

2.6.4 Forecasting Probabilístico

Existen diferencias entre Forecasting vs. Forecasting Probabilístico. En determinista se trata de predecir un valor puntual, en cambio el probabilístico predice un rango de valores posibles. Estas predicciones tienen un error, asumiendo que los errores futuros serán similares a los errores pasados, esta técnica simula diferentes predicciones tomando muestras de los errores vistos previamente en el pasado (es decir, los residuos) y agregándolos a las predicciones (Figura 21).

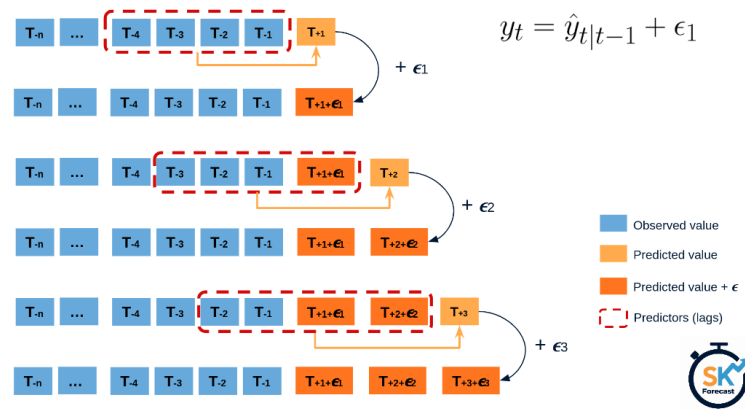


Ilustración 21. Forecasting Probabilístico. Fuente: [Skforecast](#)

Forecasting probabilístico se utiliza en contextos donde las decisiones deben considerar riesgos y márgenes de error, por ejemplo, en la planificación de la demanda, energía, logística, finanzas o meteorología. Existen tres métodos de predicción probabilística en SKForecaster, sin embargo, en esta investigación se abordará los dos primeros: Booststrapped residuals; Conformal prediction y Quantile regression.

2.6.4.1 *Booststrapped residuales*

El método de *Bootstrapped Residuals* genera intervalos de predicción mediante el re-muestreo de residuos del modelo, que consiste en generar múltiples simulaciones del futuro agregando residuos aleatorios a las predicciones puntuales. Estos residuos se originan en el entrenamiento del modelo con `fit()` que guarda los errores con el parámetro `use_in_sample_residuals=True`. A partir de estas simulaciones, se extraen los percentiles deseados para construir el intervalo, por ejemplo 80% o 95% (Figura 22).

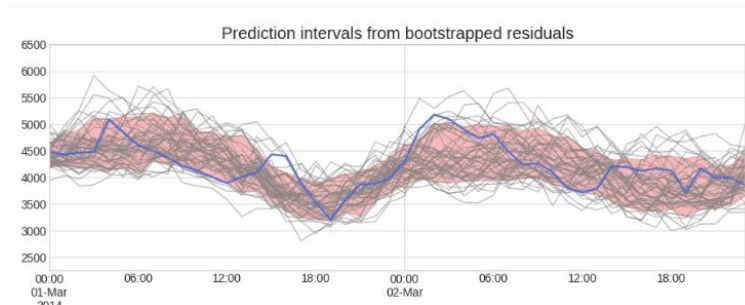


Ilustración 22. Booststrapped residuales. Fuente: [Skforecast](#)

Este método solo requiere un solo entrenamiento, lo que lo hace computacionalmente eficiente de baja a moderada. Si los errores pasados no reflejan adecuadamente la variabilidad futura, el método puede producir intervalos poco fiables. Para generar intervalos de predicción, se puede emplear tres tipos de residuos (Figura 23):

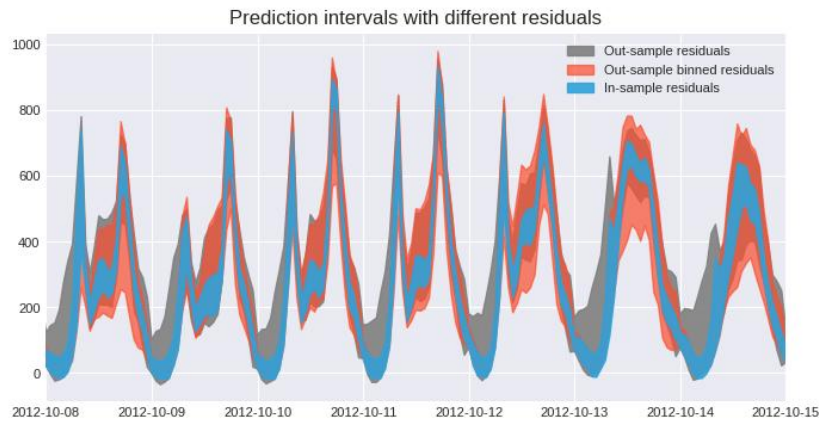


Ilustración 23. Intervalos de predicción con diferentes residuales. **Fuente:** [cienciadedatos](https://cienciadedatos.com)

Residuos In-sample: Son residuos calculados sobre los mismos datos que se usaron para entrenar el modelo. Representan la diferencia entre los valores reales y predicciones dentro del entrenamiento $e_i = |y_i - \hat{y}_i|$. Su limitación es basarse en datos ya vistos, estas subestiman el error futuro. `use_in_sample_residuals=True`

Residuos Out-of-sample (no condicionados): Se obtienen a partir de predicciones sobre datos no vistos (validación o backtesting) durante el entrenamiento. Este enfoque mejora la estimación de incertidumbre frente a los *in-sample*, aunque como limitación no captura variaciones de error dependientes de la magnitud. `use_in_sample_residuals=False`

Residuos *out-of-sample binned* (condicionados a valores predichos): Se calculan también con datos no vistos, pero se agrupan (bins) según rangos del valor predicho. Esto permite que los intervalos se ajusten dependiendo de la magnitud de la predicción, capturando mejor la variabilidad cuando existe heterocedasticidad, para estimar de forma fiable la variabilidad en cada grupo. `use_in_sample_residuals=False`

2.6.4.2 Conformal predictions y calibration

Conformal Prediction se basa en combinar las predicciones deterministas con sus residuales históricos (diferencias entre predicciones previas y valores reales). Estos residuales estiman la incertidumbre en la predicción y ajustan la amplitud del intervalo alrededor de la predicción puntual. Este método posee un procedimiento que calibra los intervalos -generados por otras técnicas: regresión cuantílica o *bootstrapping*- para alcanzar una cobertura objetivo (e.g. 80% o 95%) usando la distribución empírica de errores en datos no vistos (calibración) (Figura 24). Proceso:

1. Los intervalos de predicción se estiman para el conjunto de validación (bootstrapping). `use_in_sample_residuals=False`
2. Utiliza intervalos previstos y valores reales del conjunto de validación, el `ConformalIntervalCalibrator` aprende el factor de corrección para calibrar.
3. Los intervalos de predicción para el conjunto de prueba se ajustan utilizando el factor de corrección aprendido del conjunto de validación.

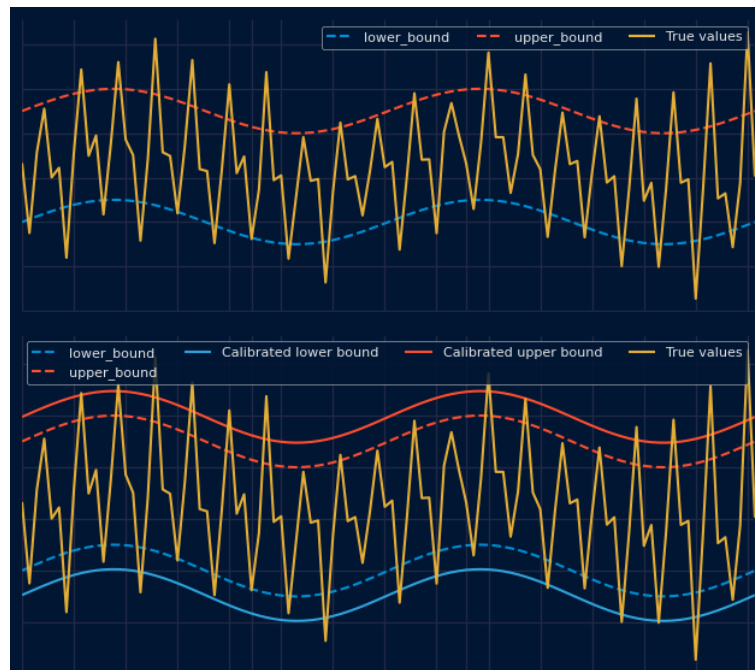
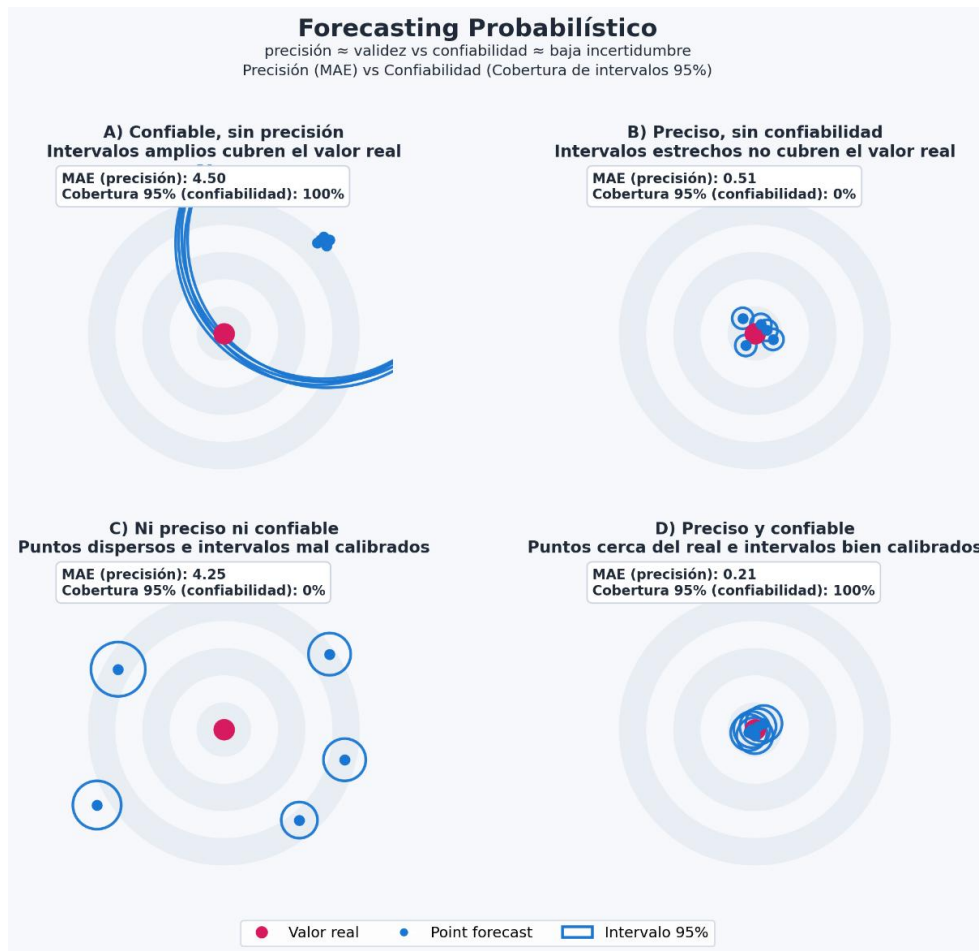


Ilustración 24. Antes y después del transformador `ConformalIntervalCalibrator`. Fuente: [Skforecast](#)

2.6.4.3 Forecasting Probabilístico: precisión, confiabilidad y trade-off

El forecasting probabilístico busca equilibrar el *trade-off* entre la precisión de los pronósticos puntuales y la confiabilidad de los intervalos de predicción, representando adecuadamente la incertidumbre asociada a cada estimación (Gneiting & Katzfuss, 2014). Forecasting probabilístico no se trata sólo de predecir bien (precisión), sino de saber cuán seguro se está al predecir (confiabilidad). El mejor modelo balancea ambos.

Intervalos más amplios suelen garantizar mayor cobertura del valor real, pero pierden poder informativo para la toma de decisiones. En cambio, intervalos estrechos resultan más específicos y útiles, aunque con menor probabilidad de contener el valor verdadero. Asimismo, es fundamental diferenciar entre precisión y confiabilidad. La precisión (análoga a la validez) se refiere a la cercanía de los pronósticos puntuales respecto al valor observado y suele evaluarse con métricas de error.



2.7 Métricas en Forecasting

En el aprendizaje automático, la evaluación del rendimiento de los modelos constituye una fase crítica para garantizar robustez, utilidad y capacidad de generalización (Hewamalage *et al.*, 2022). Las métricas no solo cuantifican la calidad de las predicciones, sino orientan las decisiones durante el ajuste de hiperparámetros y la comparación entre algoritmos. En regresión, suelen utilizarse MAE o RMSE.

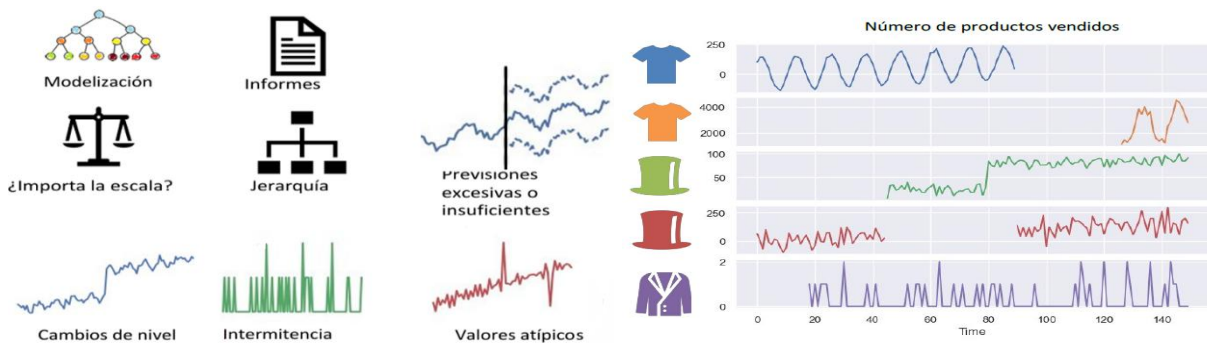


Ilustración 25. Características frecuentes que puede dificultar el pronóstico. Fuente: [Kishan Manani](#)

Las series temporales pueden presentar múltiples características que afectan la calidad de la modelización y la evaluación de pronósticos (Figura 25). Entre estas se incluyen tendencia, estacionalidad, datos no disponibles, cambios de nivel o rupturas estructurales, comportamientos intermitentes con presencia frecuente de ceros, y la aparición de valores atípicos o picos inesperados. También pueden influir factores como la escala de los datos, la estructura jerárquica de series relacionadas, y la necesidad de generar informes comparables. Según Hewamalage *et al.* (2022), estas condiciones —no estacionariedad, intermitencia o falta de normalidad— hacen que muchas métricas convencionales de error se vuelvan inadecuadas, por lo que es necesario adoptar enfoques de evaluación y modelos adaptados a las particularidades de cada serie.

En el análisis exploratorio de la serie temporal cuando revela un comportamiento intermitente, caracterizado por la presencia predominante de ceros intercalados. Este patrón de datos común en demandas irregulares o eventos discreto, complica el proceso de modelado al romper supuestos clásicos estadísticos de estacionalidad o normalidad. Hewamalage *et al.* (2022) señalan que la intermitencia, junto con la no estacionariedad y la no normalidad, son factores críticos que degradan la validez de muchas métricas convencionales, y requieren otros enfoques para la evaluación del rendimiento (p. 792). Según Hewamalage *et al.* (2022), el uso de métricas inapropiadas puede llevar a conclusiones engañosas, aparentando que ciertos métodos son competitivos cuando en realidad no lo son. El artículo advierte que es frecuente observar en la literatura científica la aplicación de métricas erróneas —por ejemplo, el MAPE sobre series con valores cercanos a cero— lo que distorsiona completamente la evaluación de modelos (p. 790). Por ello, resulta esencial seleccionar métricas adecuadas al tipo de datos y objetivo de predicción, considerando otras alternativas como msMAPE, ND, bias o escalas relativas.

2.7.1 Cálculo de errores y Cálculo de métricas de error: Separación Conceptual en la Evaluación de Modelos Forecasting en Series Temporales

En la evaluación de modelos de series temporales, Hewamalage *et al.* (2022) distinguen claramente entre dos etapas fundamentales: Error Calculation y Error Measure Calculation. La primera se refiere al cálculo individual de los errores para cada observación en el conjunto de prueba, la diferencia entre el valor real y el valor predicho en un punto temporal t , es decir, $e_t = y_t - \hat{y}_t$. Esta etapa produce una secuencia de errores crudos, que pueden adoptar diversas formas: absolutos, relativos o escalados.

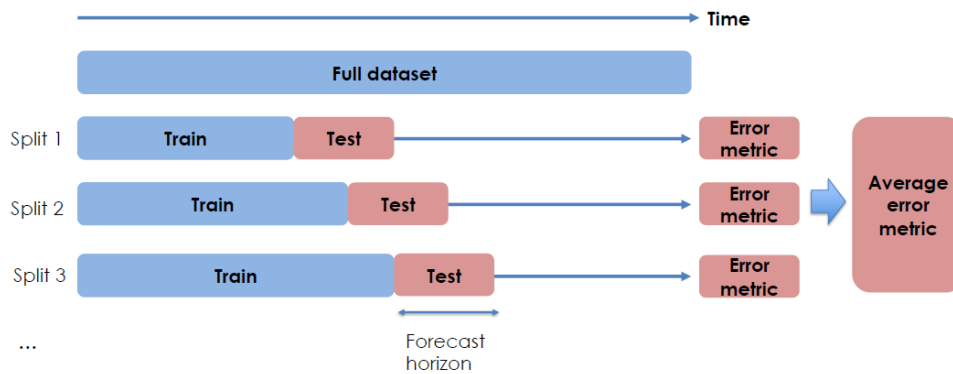


Ilustración 26. Error Vs. Métrica. Fuente: [Kishan Manani](#)

Como parte de las buenas prácticas (Figura 26), se recomienda graficar los errores individuales como parte del análisis de Error Calculation, antes de resumirlos con métricas (Error Measure Calculation). Esta información granular permite diagnosticar con precisión cómo y cuándo el modelo falla, y es útil para identificar patrones, tendencias, asimetrías o *outliers* en el comportamiento de los errores. Por otro lado, la Error Measure Calculation consiste en aplicar funciones de agregación sobre los errores individuales para obtener métricas globales de desempeño del modelo, como MAE, RMSE, MASE, entre otras. Según los autores Hewamalage *et al.* (2022), una buena práctica es mantener una separación conceptual y operativa entre estas dos etapas, evitando errores comunes como el uso indebido de estadísticas futuras al calcular errores escalados (lo que conduciría a *data leakage*) (Tabla 8).

Tabla 8. Error Calculation Vs. Error Measure Calculation

Concepto	¿Qué hace?	Resultado
Cálculo de errores (<i>Error Calculation</i>)	Genera los errores individuales para cada t	Vector de errores e_t
Cálculo de la medida del error (<i>Error Measure Calculation</i>)	Resume los errores individuales en una métrica global (e.g.: MAE, RMSE, ...)	Valor único o conjunto de métricas

2.7.2 Niveles de agregación de error en la evaluación de series temporales

En el proceso de evaluación de modelos de pronóstico, la forma en que se agregan a los errores individuales tiene un impacto en la interpretación de las métricas. *Out-of-sample error* (OOS) se refiere a los errores calculados sobre el conjunto de prueba (*test*) o validación (*validation*), es decir, valores no vistos por el modelo durante el entrenamiento. En cambio, *in-sample per series*, no es un tipo de error, sino una estrategia de escalado/normalizador del error. Para mayor comprensión, se detalla un ejemplo simple:

fullDataset = [10, 12, 11, 9, 8, 10, 12, 8]

- $y_{\text{train}} = [10, 12, 11, 9, 8]$ # datos usados (muestra) para entrenar el modelo
- $y_{\text{test}} = [10, 12, 8]$ # valores reales fuera de muestra
- $y_{\text{pred}} = [9, 11, 10]$ # predicciones fuera de muestra

Out-of-sample error: Con una métrica se calcula el error sobre el conjunto de prueba (*test*) o en el conjunto de validación (*validation*).

$$\text{RMSE}_{\text{oos}} = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2}$$
$$\text{RMSE}_{\text{oos}} = \sqrt{\frac{(10 - 9)^2 + (12 - 11)^2 + (8 - 10)^2}{3}} = \sqrt{\frac{1 + 1 + 4}{3}} = \sqrt{2} \approx \mathbf{1.41}$$

RMSE=1.41 es el error real del modelo en el *test*, lo que se reporta como desempeño.

In-sample per series: Cálculo del promedio en la muestra como base para escalar error:

$$\overline{y^{\text{in}}} = \frac{1}{T} \sum_{t=1}^T y_t$$
$$\text{Promedio in-sample} = \frac{10 + 12 + 11 + 9 + 8}{5} = 10$$

Ahora se calcula escalador sMSE (scaled MSE):

$$\text{sMSE} = \frac{1}{n} \sum_{t=1}^n \left(\frac{y_t^{\text{oos}} - \hat{y}_t^{\text{oos}}}{\overline{y^{\text{in}}}} \right)^2$$
$$\text{sMSE} = \frac{1}{3} \left(\left(\frac{10 - 9}{10} \right)^2 + \left(\frac{12 - 11}{10} \right)^2 + \left(\frac{8 - 10}{10} \right)^2 \right) = \frac{1}{3} (0.01 + 0.01 + 0.04) = \frac{0.06}{3} = \mathbf{0.02}$$

sMSE=0.02 representa un 2% del cuadrado del valor promedio de la serie, en otras palabras, el modelo predice con gran precisión con un error muy bajo.

Si el valor es **0.5**, el error sigue siendo razonable: representa el **50 %** del nivel cuadrado promedio, lo cual refleja un **desempeño aceptable**. Un valor de **1.0** sugiere que el error es tan grande como el promedio cuadrado de la serie, señalando que el modelo **no mejora sobre una predicción simple promedio**. Un valor de ≥ 1.0 indica que el modelo está **cometiendo errores grandes** con relación al tamaño de los datos, es decir, el modelo tiene un desempeño deficiente. Esta interpretación hace del sMSE una herramienta útil para comparar modelos bajo una escala común y comprensible.

2.7.3 Métrica dependiente de la escala: Root Median Squared Error (RMdSE)

Evalúa el desempeño de un modelo mediante la raíz cuadrada de la mediana de los errores al cuadrado. A diferencia del RMSE, que utiliza la media y es sensible a valores atípicos, el RMdSE emplea la mediana como medida de tendencia central, lo que lo hace más robusto frente a *outliers*. Esta métrica resulta especialmente útil en contextos donde existen observaciones atípicas o distribuciones asimétricas del error, como ocurre con series intermitentes o datos con alta dispersión. Matemáticamente, se define como:

Ecuación 1

$$\text{RMdSE} = \sqrt{\text{median}((y_t - \hat{y}_t)^2)}$$

Donde:

- y_t : valor real
- \hat{y}_t : valor predicho

Desde el punto de vista interpretativo, el RMdSE al no estar escalado, sus valores son dependientes de la escala de la serie y se expresan en las mismas unidades que la variable pronosticada, lo que facilita su interpretación directa. Un valor bajo indica que la mayoría de las predicciones se encuentran cerca de los valores reales.

2.7.4 Métrica basada en porcentajes de error: sMSE (Scaled Mean Squared Error)

La métrica sMSE (Scaled Mean Squared Error) es una variante escalada del error cuadrático medio que permite evaluar el desempeño de un modelo en relación con el nivel promedio de la serie temporal. A diferencia del RMSE, que puede ser difícil de interpretar cuando se comparan series con diferentes magnitudes, el sMSE proporciona una forma de normalización que facilita la comparación entre modelos y entre series.

Este enfoque no escala por desviación estándar ni por varianza, sino por el promedio in-sample de los valores reales, lo cual evita la inestabilidad que puede surgir al dividir por varianzas pequeñas. Gracias a esta característica, resulta más estable en ciertos contextos, especialmente en datos con baja variabilidad o donde se busca una métrica robusta. En conjunto, el sMSE ofrece una evaluación escalada que es conceptualmente más sólida y comparativamente más robusta que el RMSE directo, siendo recomendado en la literatura como una alternativa útil cuando la varianza no es una opción fiable de escalado (Hewamalage *et al.*, 2023; Petropoulos & Kourentzes, 2015).

Ecuación 2

$$sMSE = \frac{1}{n} \sum_{t=1}^n \left(\frac{y_t - \hat{y}_t}{\frac{1}{T} \sum_{t=1}^T y_t} \right)^2$$

Donde:

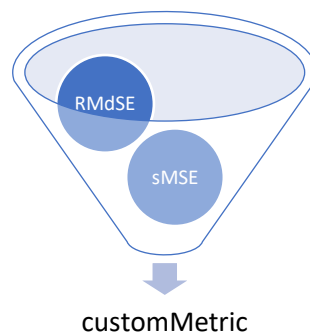
- y_t : valor real
- \hat{y}_t : valor predicho
- $y_t - \hat{y}_t$: error de pronóstico
- $\frac{1}{T} \sum_{t=1}^T y_t$: media *in-sample* de los valores reales
- T : tamaño del conjunto *in-sample* (puede coincidir con n)

Un valor de sMSE cercano a cero indica predicciones muy precisas; entre 0.01 y 0.1 sugiere un buen desempeño con errores pequeños; valores entre 0.1 y 1.0 reflejan errores moderados; mientras que un sMSE superior a 1.0 indica que el error cuadrático medio supera el valor medio de la serie, lo que señala un ajuste deficiente del modelo.

2.7.5 Combinación de métricas

Combinar métricas como RMdSE (dependiente de escala) y sMSE (normalizada) puede ser adecuado siempre que ambas compartan una base común (mínimo en 0, error cuadrático). Esta combinación permite capturar tanto el error absoluto como el error relativo, ofreciendo una evaluación equilibrada del modelo. Esta estrategia requiere cautela en contextos de múltiples series o distintas escalas.

En contextos de este trabajo forecasting univariado, la combinación ponderada de métricas absolutas y relativas puede enriquecer la evaluación del modelo. En este trabajo, se emplea una métrica compuesta basada en RMdSE y sMSE, asignando igual peso a ambas ($\alpha = 0.5$), con el objetivo de capturar tanto la precisión en escala original como la estabilidad relativa respecto a la media histórica del sistema.



En este estudio, la selección de métricas de evaluación se basó en criterios de Hewamalage *et al.* (2023) frente a características específicas de la serie, como la intermitencia y presencia de *outliers* (Tabla 9 y 10).

Tabla 9. Métricas apropiadas a este caso de estudio, basado en Hewamalage *et al.* (2023)

Métrica	Justificación
RMdSE	✓ Robusta ante outliers e intermitencia (usa mediana).
msMAPE	✓ Alternativa mejor que MAPE para datos con ceros/intermitentes
sMSE	✓ Escala el error por media in-sample → adecuado para evitar divisores pequeños (intermitente).

Tabla 10. Métricas excluidas a este caso de estudio, basado en Hewamalage *et al.* (2023)

Métrica	Justificación
RMSE	✗ Sensible a outliers y heterocedasticidad.
MAE	✗ No adecuada para intermitencia ni outliers significativos.
MAPE	✗ No definida con ceros; se distorsiona con valores pequeños (tu serie va de 0 a 10).
sMAPE	✗ Puede ser inestable con valores pequeños o ceros.

2.8 Selección del modelo, herramienta, técnica y enfoque

Los modelos estadísticos y econométricos, como ARIMA, SARIMA y VAR, son ampliamente utilizados en el pronóstico meteorológico y marítimo por su capacidad para representar relaciones temporales con base teórica. Sin embargo, su aplicabilidad puede verse limitada en escenarios con relaciones no lineales complejas o grandes volúmenes de datos.

Skforecast resulta útil cuando se trabaja con estaciones meteorológicas, ya que permite aplicar modelos de *machine learning* con facilidad, incorporar variables exógenas (como viento, marea, visibilidad), validar mediante *backtesting*, y actualizar las predicciones sin reentrenar todo el modelo. Su estructura modular facilita un flujo reproducible y transparente. En contraste, las redes recurrentes implementadas en PyTorch son adecuadas cuando se requiere modelar patrones secuenciales complejos o dependencias cruzadas entre múltiples sensores u observaciones multivariadas, por ejemplo, la interacción entre oleaje, viento y corrientes marinas. No obstante, estas redes requieren mayor procesamiento, diseño arquitectónico, y son menos interpretables.

Skforecast es recomendable para aplicaciones operativas y de diagnóstico, al combinar modelos de *machine learning* con interpretabilidad y flexibilidad. PyTorch, con redes

recurrentes, resulta más adecuado para desarrollos experimentales y contextos con datos abundantes y patrones complejos. En contraste, los modelos estadísticos y econométricos, aunque útiles para análisis explicativos, son menos eficaces ante relaciones no lineales o grandes volúmenes de datos, lo que limita su aplicabilidad en escenarios modernos.

2.8.1 Selección del modelo acorde a la metodología KDD

La metodología KDD (Knowledge Discovery in Databases) tiene por propósito la extracción de patrones relevantes y conocimiento a partir de datos. Los modelos de *machine learning* por medio de Skforecast ofrecen una solución equilibrada entre precisión predictiva y capacidad de incorporar múltiples variables con una razonable interpretabilidad de los resultados. A diferencia de los modelos *deep learning* y modelos estadísticos clásicos —e.g. ARIMA o SARIMA— que dependen de supuestos formales y estructuras rígidas. Los enfoques de ML permiten capturar relaciones no lineales y patrones complejos sin requerir los supuestos estadísticos como estacionaria, varianza homogénea, etc. Los modelos de ML son reproducibles y apropiados para contextos donde la explicabilidad y la trazabilidad del conocimiento son prioritarias en estudios hidrometeorológicos o logísticos. Esta elección responde al propósito de KDD, que es extraer conocimiento útil, interpretable y validado empíricamente para la comprensión del fenómeno bajo estudio, más allá del despliegue en producción (Figura 27).

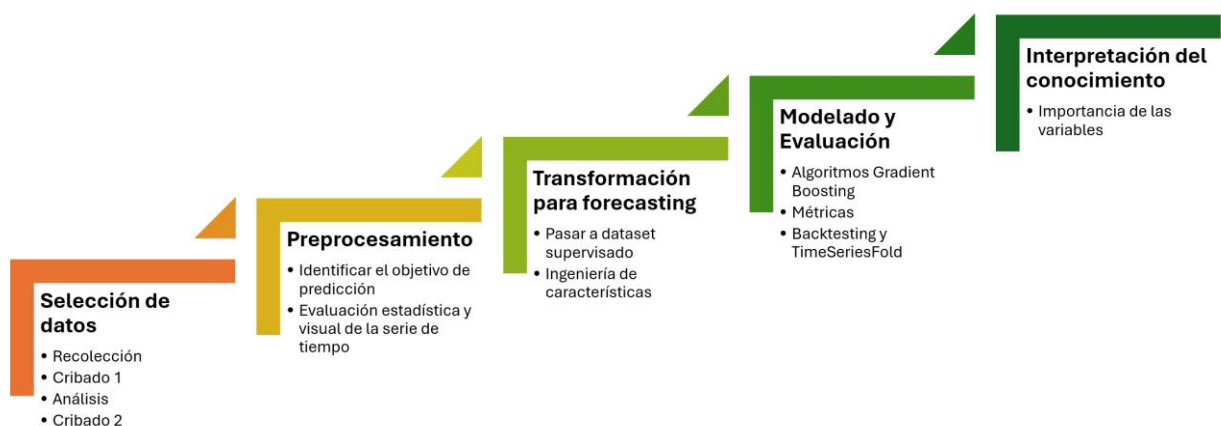


Ilustración 27. KDD aplicado al tema de investigación

2.8.2 Selección de la herramienta para el modelado con *machine learning*

Existen múltiples librerías orientadas a Forecasting, tales como Functime, SKTime, Darts, Nixtla. Para este trabajo se explorará SKForecast, opción que facilita el desarrollo de prototipos, así como el despliegue en producción.

Skforecast cuenta con una comunidad activa en crecimiento. Aunque no es tan amplia como la de librerías más consolidadas como Prophet o XGBoost, se caracteriza por ser técnica, receptiva y estar bien documentada, por lo que es valioso para proyectos académicos o industriales que demandan estabilidad y transparencia. Ventajas:

1. Estrategias flexibles de forecasting: permite construir pronósticos de múltiples pasos (multi-step) con control total del horizonte.
2. Compatibilidad con modelos avanzados: integra cualquier modelo compatible con scikit-learn: RandomForest, GradientBoosting, Ridge, etc. Además, de modelos externos como XGBoost, LightGBM, CatBoost, HistGradientBoosting.
3. Validación temporal robusta: *backtesting*, *walk-forward* y Evaluación multihorizonte. Usa métricas estándar: MAE, RMSE, R^2 , msMAPE, F1-Binary, etc
4. Soporte para variables exógenas: conserva consistencia del entrenamiento y predicción con exógenas. Permite agregar múltiples regresores externos como meteorología, precio, indicadores económicos.
5. Automatización de lags y ventanas: simplifica la ingeniería de características temporales. Genera automáticamente variables *lags* o características de ventana (rolling_mean, std, etc.)
6. Actualización sin reentrenar: permite actualizar modelos entrenados con nuevos datos sin reiniciar el entrenamiento, gracias a last_window. Útil para flujos productivos o streaming de datos.
7. Interpretabilidad: incluye funciones para calcular e interpretar (*feature importance*). Mide cuánto influye cada variable -o lag- en la predicción del modelo.
8. Ajuste de hiperparámetros: compatible con GridSearchCV, RandomSearchCV y Optuna para tuning automatizado.
9. Diseño modular y reproducible: su API está orientada a reproducibilidad y escalabilidad. Compatible con Pipelines, integración con notebooks, y estructuras de experimentación.
10. Código abierto y documentación completa: licencia MIT uso libre académico e industrial. Documentación oficial muy clara: <https://skforecast.org>

2.8.3 Selección de enfoque acorde a la herramienta y capacidad de computo

Se adoptó un enfoque **univariante** y **multistep** para el modelado de la serie temporal, centrado exclusivamente en la variable objetivo (Figura 28). El enfoque multistep, por su parte, permite anticipar varios pasos hacia adelante en un solo proceso de inferencia.

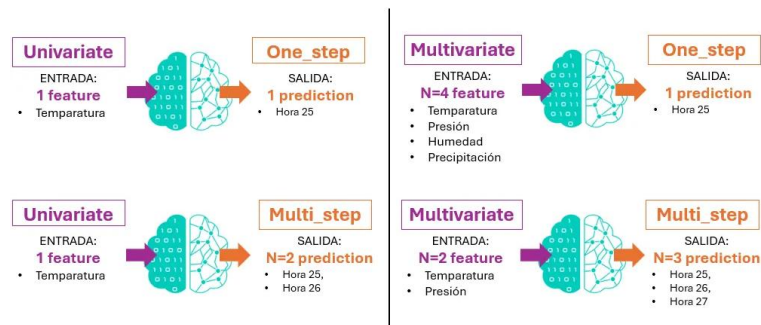


Ilustración 28. Tipos de enfoques en *Forecasting*

La estrategia planteada busca equilibrar la fiabilidad del error y la eficiencia en tiempo y memoria, priorizando un esquema que, sin ser el más costoso, mantiene una precisión aceptable en el horizonte de forecast (Figura 29). Para ello se empleó *ForecasterRecursive*, adecuado en modelos de árboles de decisión para series temporales, pues permite predicciones recursivas con menor carga computacional que enfoques como *ForecasterRecursiveMultiSeries* o *ForecasterDirect*.



Ilustración 29. Fiabilidad Vs. Tiempo y memoria. **Fuente:** Adaptado de [Kishan Manani](#)

A partir de mayo de 2025 el desarrollo y la ejecución de los modelos se llevaron a cabo en un equipo equipado con un procesador de 13th Gen Intel Core i7-13620H, que cuenta con 16 núcleos físicos y una frecuencia base de 2.92 GHz, acompañado de 15.64 GB de memoria RAM. Para tareas que requieren procesamiento paralelo, se contó con GPU NVIDIA GeForce RTX 4050 con 6 GB de memoria dedicada (6141 MB) y soporte CUDA versión 12.7, permitiendo manejar “medianamente” cargas computacionales asociadas al entrenamiento y validación en los modelos de *machine learning* planteados.

CAPÍTULO 3

3. RESULTADOS Y ANÁLISIS

Este capítulo aborda la implementación de la metodología Knowledge Discovery in Databases (KDD) y el análisis de costos de la solución.

3.1 Problema

El propósito de esta investigación es sobre el pronóstico de la demanda de remolcadores para buques en un terminal portuario de hidrocarburos. En la literatura científica no se encuentran este tipo de casos, no obstante, se toma el reto de la investigación e innovación en este amplio espectro del *forecasting*.

3.2 FASE 1: Selección de datos

3.2.1 Recolección

Los datos (ver sección 1.7) para el estudio fueron proporcionados por la fuente bajo un acuerdo de confidencialidad y anonimato.

Se trabajó en los *dataset*: remolcadores (34'657 registros desde 2016 mayo del 2024) y buques (26'945 registros desde 1999 hasta mayo del 2024). Se ejecuta el análisis de la consistencia de datos, con varias acciones, tales como: suprimir filas y columnas duplicadas o irrelevantes; detección de valores nulos y vacíos; imputación; limpiar de errores tipográficos y letras no reconocidas; renombrar y reubicar columnas; asignar el tipo de dato correcto; agregar columnas para incorporar información relevante -e.g. relación entre variables, categorización-, corroborar datos; entre otras.

3.2.2 Primer cribado

A partir de los dos *dataset* pretratados: buques y remolcadores. Estos se unifican quedando un *dataset* (Tabla 11) con registros individuales de la actividad del remolcador, esto quiere decir, un buque solicita una maniobra, cada maniobra involucra n remolcadores, por lo tanto, existen n registros. Por ejemplo, en la Tabla 12 tenemos, la instancia de un buque -llegada 31ene2024 y salida 1feb2024-, en el caso de la llegada del buque, solicitó la maniobra atraque que involucró 3 remolcadores, por lo tanto, se tiene 3 registros.

Tabla 11. Dataset unificado entre buque y remolcadores

	ID	TIPO_MANIOBRA	DATETIME_INI	DATETIME_FIN	HORAS_TOTAL_x
0	26857	Desatraque	2024-02-01 06:30:00	2024-02-01 08:30:00	2.00
1	26844	Atraque	2024-02-01 08:30:00	2024-02-01 11:00:00	2.50
2	27022	Atraque	2024-04-01 06:00:00	2024-04-01 09:00:00	3.00
3	27112	Atraque	2024-05-01 05:30:00	2024-05-01 07:30:00	2.00
4	27097	Atraque	2024-05-01 09:30:00	2024-05-01 12:00:00	2.50
...
33068	19113	Desatraque	2016-12-27 19:25:00	2016-12-27 21:15:00	1.83
33069	19279	Desatraque	2017-01-28 13:40:00	2017-01-28 15:20:00	1.67
33070	19474	Desatraque	2017-04-28 09:15:00	2017-04-28 10:50:00	1.58
33071	19198	Atraque	2016-12-29 17:30:00	2016-12-29 18:50:00	1.33
33072	19403	Atraque	2017-03-30 17:20:00	2017-03-30 19:20:00	2.00

33073 rows × 5 columns

Tabla 12. Registros de remolcadores en la instancia de un buque

	ID	TIPO_MANIOBRA	DATETIME_INI	DATETIME_FIN	HORAS_TOTAL_x
0	26857	Desatraque	2024-02-01 06:30:00	2024-02-01 08:30:00	2.0
633	26857	Atraque	2024-01-31 15:30:00	2024-01-31 18:00:00	2.5
10533	26857	Atraque	2024-01-31 15:30:00	2024-01-31 18:00:00	2.5
22249	26857	Desatraque	2024-02-01 06:30:00	2024-02-01 08:30:00	2.0
23998	26857	Atraque	2024-01-31 15:30:00	2024-01-31 18:00:00	2.5

Para obtener la suma de remolcadores por cada maniobra solicitada indistinto del buque. Se agrupa los registros por maniobra, agregando la columna ‘N_REMOLCADOR_xMANIOBRA’, de esta forma se obtiene en la Tabla 13, el número de remolcadores por cada maniobra solicitada, adjuntando el tipo de maniobra y las características de los buques —capacidad de tonelaje, calado, eslora y tipo de nave—. Luego, se delimita a partir del 2017 en adelante.

Tabla 13. Primer cribado del *dataset*

ID	TIPO_MANIOBRA	DATETIME_INI	DATETIME_FIN	HORAS_TOTAL_x	N_REMOLCADOR_xMANIOBRA	TRB	CALADO	RANGO_ESLORA	TIPO_NAVE
19217	Atraque	2017-01-01 10:50:00	2017-01-01 11:50:00	1.00	2	2077.0	6.60	<=150	B/T
19159	Atraque	2017-01-01 12:30:00	2017-01-01 13:55:00	1.42	2	2025.0	5.76	<=150	B/T
19159	Otros	2017-01-01 14:00:00	2017-01-01 17:00:00	3.00	1	2025.0	5.76	<=150	B/T
19159	Desatraque	2017-01-02 05:45:00	2017-01-02 06:30:00	0.75	1	2025.0	5.76	<=150	B/T
19219	Atraque	2017-01-02 06:30:00	2017-01-02 07:45:00	1.25	1	4206.0	6.60	<=150	B/T

3.2.3 Análisis Exploratorio de Datos (EDA): descriptivo

Se procedió al análisis exploratorio con el fin de conocer el comportamiento de la logística portuaria, cabe mencionar que el año 2024 consta registrados datos hasta el mes de mayo.

En el mapa de calor de la Figura 30, se detalla la frecuencia diaria de maniobras con remolcadores por año, e. g. en el 2023 hubo 9 maniobras en 54 ocasiones dentro del año, la más alta de todos los años registrados. Por otro lado, en la Figura 25 resumiendo toda la información, consta un máximo 7 maniobras al día en más de 300 ocasiones entre todos los años; también se observa la nulidad en maniobras en 60 ocasiones.

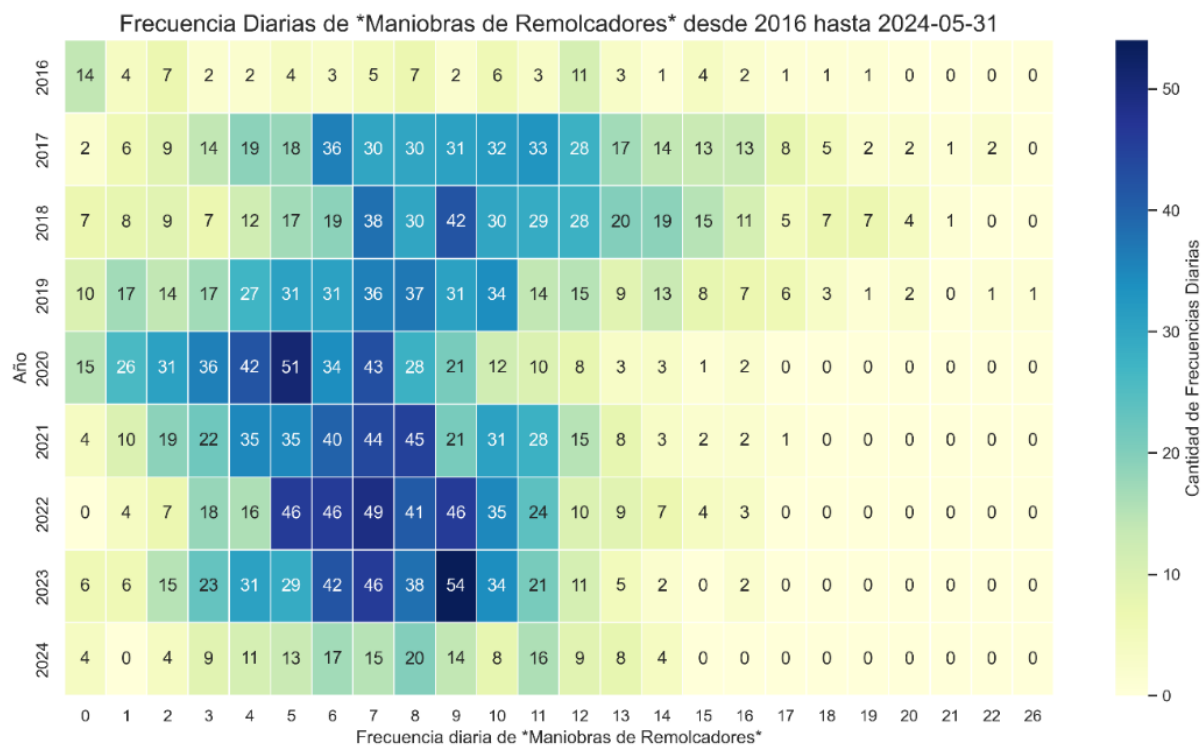


Ilustración 30. Suma de maniobras por: frecuencia diaria resumida desglosada por año

En la Figura 32, se presenta la evolución de la frecuencia diaria de maniobras de remolcadores entre enero y mayo de 2024. Se aprecia una alta varianza diaria y una leve tendencia ascendente, a partir de abril de 2024. Los valores atípicos hacia la nulidad de operaciones podrían corresponder a eventos no operativos o días no laborables.

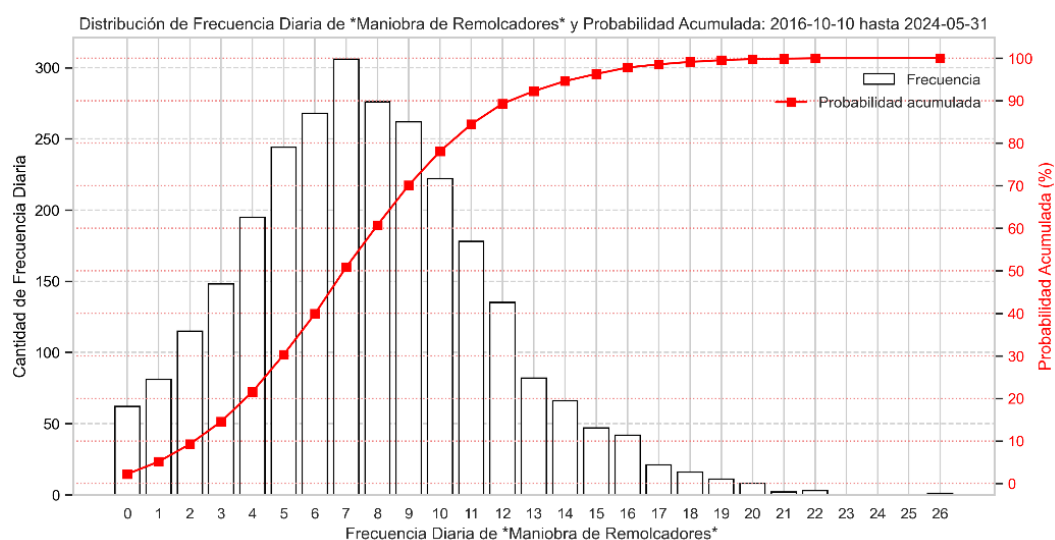


Ilustración 31. Suma de maniobras: frecuencia diaria resumida

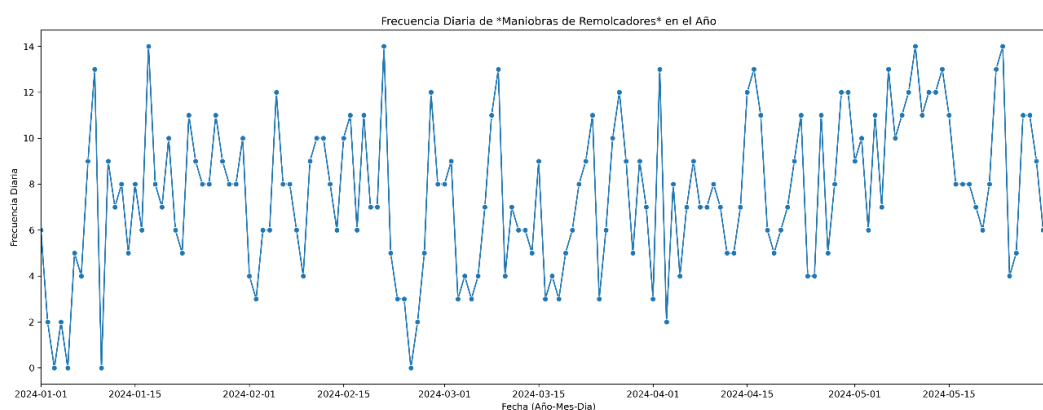


Ilustración 32. Cantidad de Maniobras con Remolcadores: frecuencia diaria en los 5 primeros meses del 2024

En las Figura 33 y 34, se detalla el número de remolcadores por maniobra de acuerdo con la eslora del buque. Aquí, la maniobra de atraque tiene un mayor número de remolcadores -3 y 4- a diferencia del desatraque.

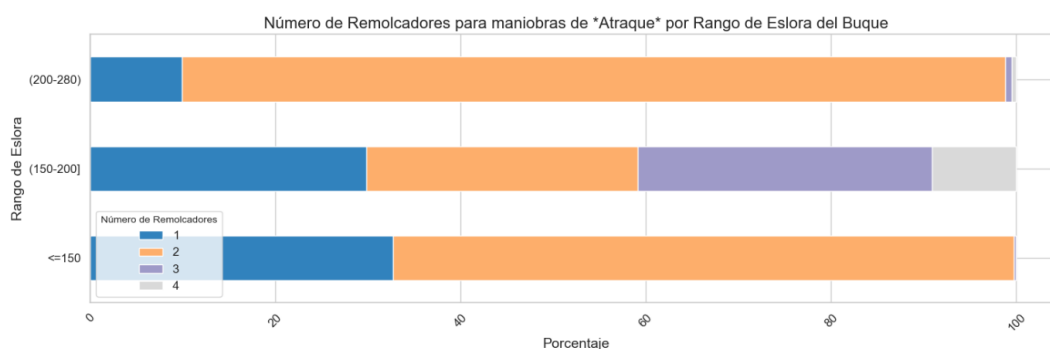


Ilustración 33. Maniobras de atraque: Número de remolcadores

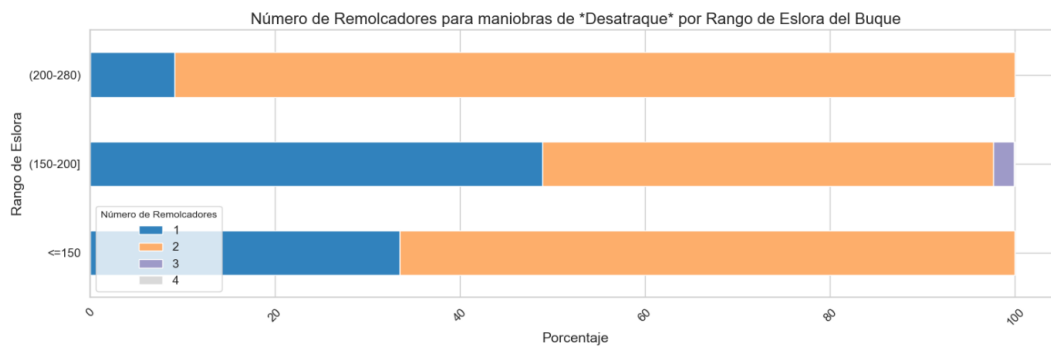


Ilustración 34. Maniobras de desatraque: Número de remolcadores

3.2.4 Segundo cribado

Luego del análisis exploratorio, se observa diferencias significativas en la interacción buques y remolcadores, entre ellas:

1. El arribo de los buques gaseros es esporádico (Figura 35)
2. La estancia de los buques gaseros en su mayoría tiende a ser una semana, por lo que puede involucrar varias maniobras con remolcadores distintas a las cotidianas: atraque y desatraque (Figura 36)
3. Los buques gaseros requieren un mayor tiempo de maniobra con remolcadores en comparación a los buques tanques (Figura 37)
4. La concentración de remolcadores está ubicada en el puerto principal, teniendo que trasladarse aproximadamente 16 millas náuticas para llegar al puerto secundario donde arriban los buques gaseros (Figura 1)

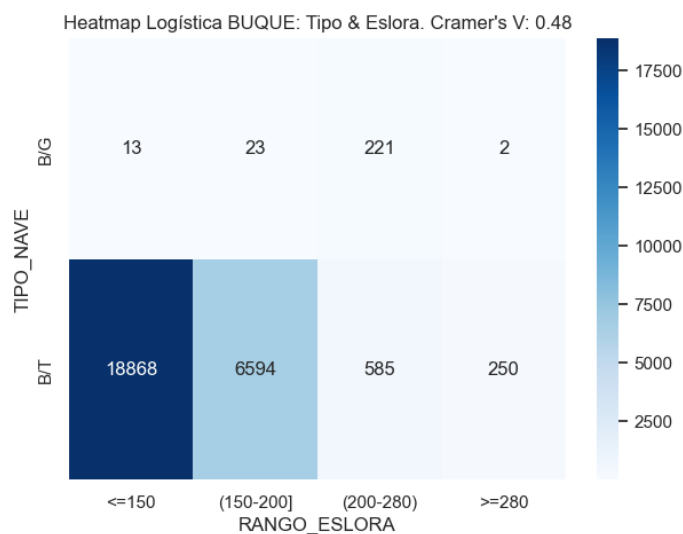


Ilustración 35. Características de buques que arriban al puerto

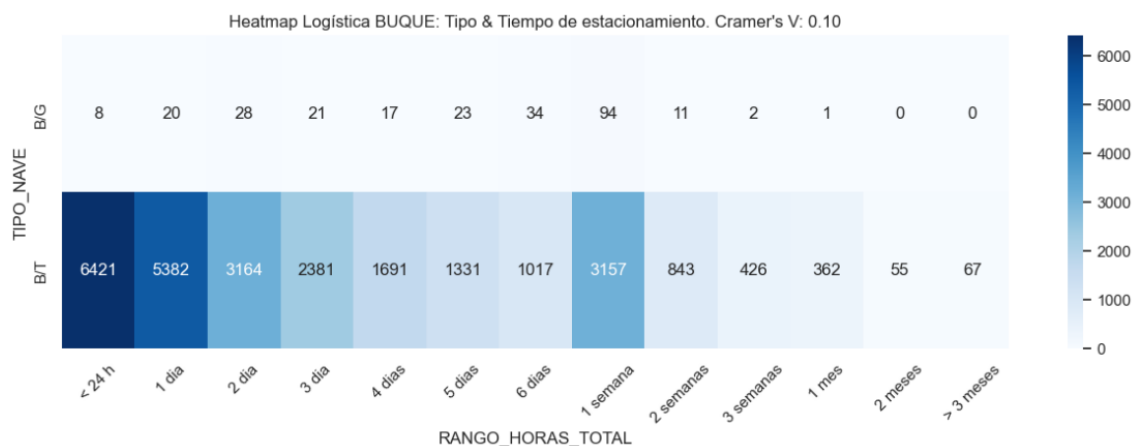


Ilustración 36. Tiempo de estadía por tipo de buque

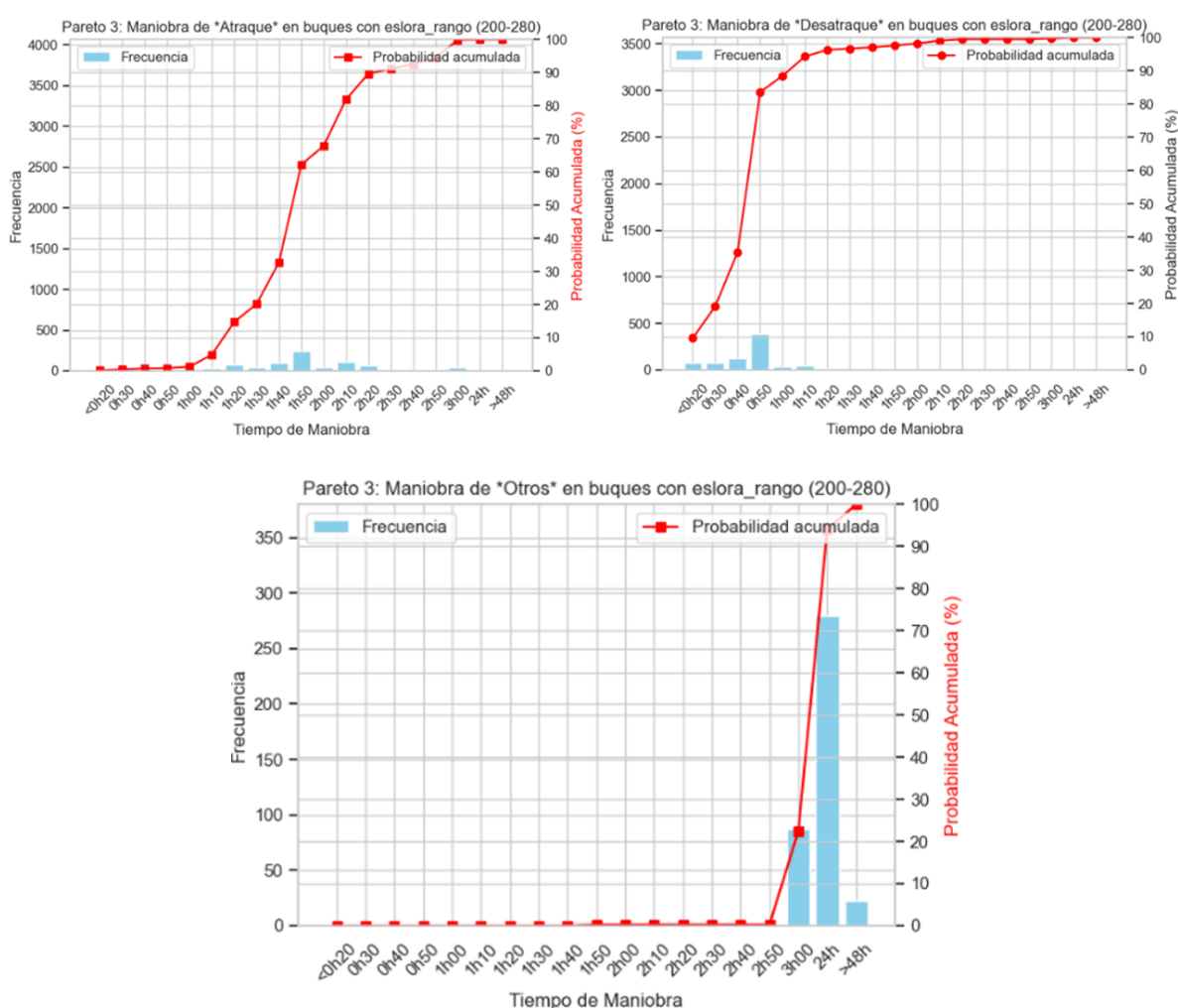


Ilustración 37. Tiempo de servicio de remolcadores con buques por tipo de maniobra y por eslora (longitud) del buque

Con este contexto, para evitar un desbalance en los datos, se decidió **sectorizar** el conjunto de datos y trabajar únicamente con maniobras de buques tanque (Figura 35). También, se delimitó la temporalidad a partir del 1 de enero de 2021, debido a la gran cantidad de registros acumulados desde 2017. En razón, de que para los algoritmos de

machine learning, un mayor volumen de datos implica mayor demanda de recursos computacionales, tanto en memoria como en procesamiento y tiempo.

3.3 FASE 2: Pre-procesamiento

3.3.1 Identificación del objeto de predicción y reestructuración del *dataset*

Para enfocar la variable objetivo 'y' con el número de remolcadores solicitados en el tiempo, se requiere una segunda transformación. Dado que el último *dataset*, los eventos de maniobra se traslapan debido al lapso de tiempo que conlleva cada maniobra. Por ejemplo, existen maniobras que pueden durar 24 horas o más, en este tiempo a la par pueden existir varias maniobras ejecutándose (Figura 37).

Esta tarea tuvo dificultades e intentos fallidos, luego de un análisis exhaustivo, se optó por aplicar una transformación temporal de eventos y acumulación de remolcadores activos.

En esta transformación, se busca conocer, minuto a minuto, cuántas maniobras están activas y cuántos remolcadores están en operación. Para lograrlo, se genera una serie temporal continua a nivel de minutos, y se marca el inicio y fin de cada evento (maniobra). A través de sumas acumuladas, se obtiene en cada momento el número de maniobras activas (*SUMA_T*) y el total de remolcadores operativos (*SUMA_N*). Esto permite crear una nueva representación temporal estructurada, que ahora sí puede ser utilizada como base para definir una **variable objetivo clara**. Por ejemplo, si se quisiera predecir cuántos remolcadores estarán activos en los próximos 15 minutos, la columna *SUMA_N* puede ser formalmente tratada como variable objetivo. Finalmente, debido a que una granularidad alta -un minuto- requiere mayor poder de cómputo, se generaliza aplicando una frecuencia de 10 minutos con el promedio o máximo número remolcadores activos (*SUMA_N*) dentro de cada intervalo. A continuación, en base a la Figura 38, el detalle:

- **t#**: Representa cada maniobra
- **SUMA_T**: Es la suma de maniobras
- **n#**: Número de remolcadores en uso por maniobra
- **SUMA_N**: La suma de remolcadores entre las maniobras
- **Frecuencia 10min_max**: Baja la granularidad de la frecuencia en lapsos de tiempo de 10 minutos, tomando el máximo o el promedio de remolcadores activos.

ENTRADA

DATETIME_INI	DATETIME_FIN	N_REMOLCADOR_xMANIOBRA
01/01/2017 10:10	01/01/2017 10:15	2
01/01/2017 10:12	01/01/2017 10:17	1
01/01/2017 10:15	01/01/2017 10:20	3
01/01/2017 10:22	01/01/2017 10:25	5

SALIDA

DATETIME	t1	t2	t3	t4	SUMA_T	n1	n2	n3	n4	SUMA_N	frecuencia
01/01/2017 10:10	1	0	0	0	1	2				2	
01/01/2017 10:11	1	0	0	0	1	2				2	
01/01/2017 10:12	1	1	0	0	2	2	1			3	
01/01/2017 10:13	1	1	0	0	2	2	1			3	
01/01/2017 10:14	1	1	0	0	2	2	1			3	
01/01/2017 10:15	0	1	1	0	2		1	3		4	4
01/01/2017 10:16	0	0	1	0	1		1	3		4	
01/01/2017 10:17	0	1	1	0	2			3		3	
01/01/2017 10:18	0	0	1	0	1			3		3	
01/01/2017 10:19	0	0	1	0	1			3		3	
01/01/2017 10:20	0	0	0	0	0					0	
01/01/2017 10:21	0	0	1	0	1					0	
01/01/2017 10:22	0	0	1	1	1				5	5	5
01/01/2017 10:23	0	0	1	1	1				5	5	
01/01/2017 10:24	0	0	1	1	1				5	5	
01/01/2017 10:25	0	0	1	0	1					0	

Ilustración 38. Restructuración del *dataset* para la variable objetivo

3.3.2 Exploración estadística y visual de la estructura temporal de la serie

La exploración estadística y visual de la estructura temporal es un paso fundamental en el análisis de series temporales, esta exploración permite comprender los patrones subyacentes y las características propias de la serie antes de aplicar cualquier modelo predictivo. Con herramientas estadísticas y visuales, es posible identificar componentes como tendencia, estacionalidad, nivel, ciclos, outliers, interrupciones, anomalías. Este análisis inicial orienta la selección de técnicas de transformación, modelado y validación más adecuadas, para garantizar que el enfoque utilizado se ajuste a la naturaleza de los datos y a los objetivos del estudio. Antes de pasar a la exploración, se visualiza el *dataset* preprocesado en la fase anterior con 10 minutos de frecuencia, uno para la demanda promedio (Figura 39) y otro para la demanda máxima (Figura 40).

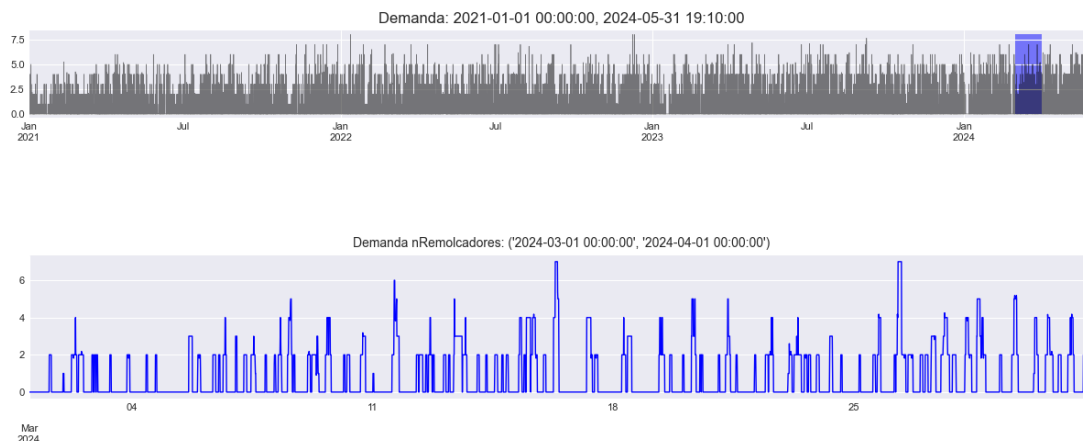


Ilustración 39. Demanda de remolcadores promedio en una frecuencia de 10 minutos

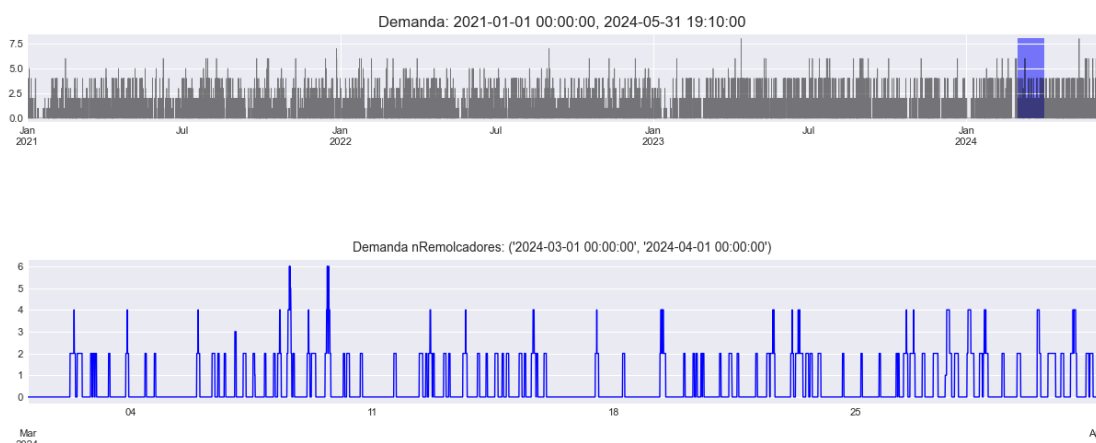


Ilustración 40. Demanda de remolcadores máximo en una frecuencia de 10 minutos

Para simplicidad en la exploración, se aplica las siguientes herramientas estadísticas y visuales sobre el *dataset* demanda promedio de remolcadores con frecuencia 10 minutos.

3.3.2.1 Normalidad

Se aplicó la prueba estadística de Kolmogorov-Smirnov (K-S), H_0 (nula): los datos siguen una distribución normal y H_1 (alternativa): los datos no siguen una distribución normal. El resultado arrojó un estadístico $D = 0.4030$ y un p -valor = 0.0000, dado que el p -valor es menor al umbral (0.05), se rechaza la hipótesis nula, concluyendo que los datos no siguen una distribución normal. Así mismo, en el gráfico QQ-plot muestra una comparación entre los cuantiles teóricos de una distribución normal (línea roja) y los valores observados (puntos azules) (Figura 41). En este caso, los puntos se desvían considerablemente de la línea recta, especialmente en los extremos, lo que indica que

los datos **no siguen una distribución normal o gaussiana**, esto refuerza los resultados de la prueba de Kolmogorov-Smirnov, confirmando visualmente la falta de normalidad.

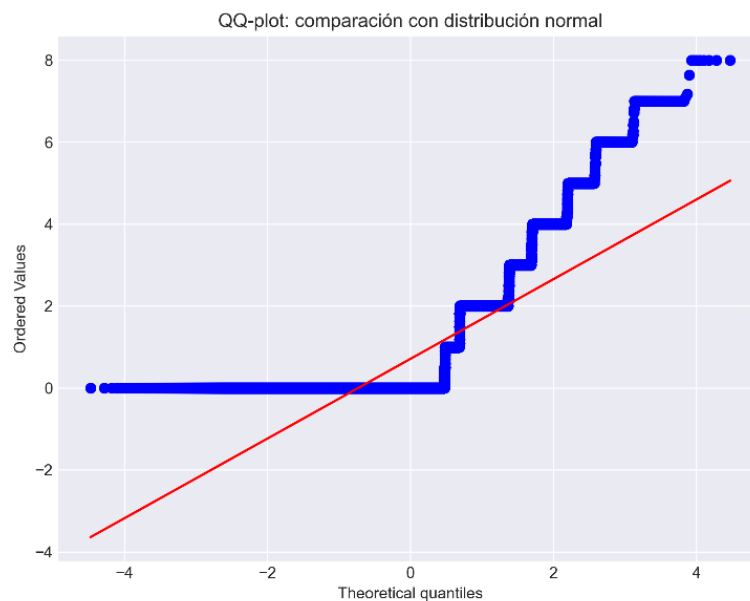


Ilustración 41. Prueba visual QQ-plot

3.3.2.2 Estacionalidad

El Figura 42 en la sección c se ubica la descomposición estacional (promedio de demanda por día), donde muestra patrones repetitivos a lo largo del tiempo, sugiriendo una estructura cíclica diaria asociada a rutinas. Este patrón es consistente con los resultados de los gráficos de distribución por mes, día de la semana y hora del día, donde se observa una variabilidad sistemática en la demanda, destacándose picos específicos durante ciertas horas o días.

Adicionalmente, los gráficos de autocorrelación (ACF) refuerzan esta observación. En particular, el de 1008 períodos (1 semana) muestra repuntes periódicos en los rezagos, lo que indica dependencia temporal con ciclos repetitivos, característica propia de series con estacionalidad. Así mismo, el de 144 períodos (1 día) refleja una estructura de autocorrelación que no decae completamente, lo cual es típico en datos con fuerte componente estacional. En conjunto, estos resultados confirman la necesidad de incorporar variables estacionales o transformaciones adecuadas en el modelado para capturar estos patrones cíclicos.

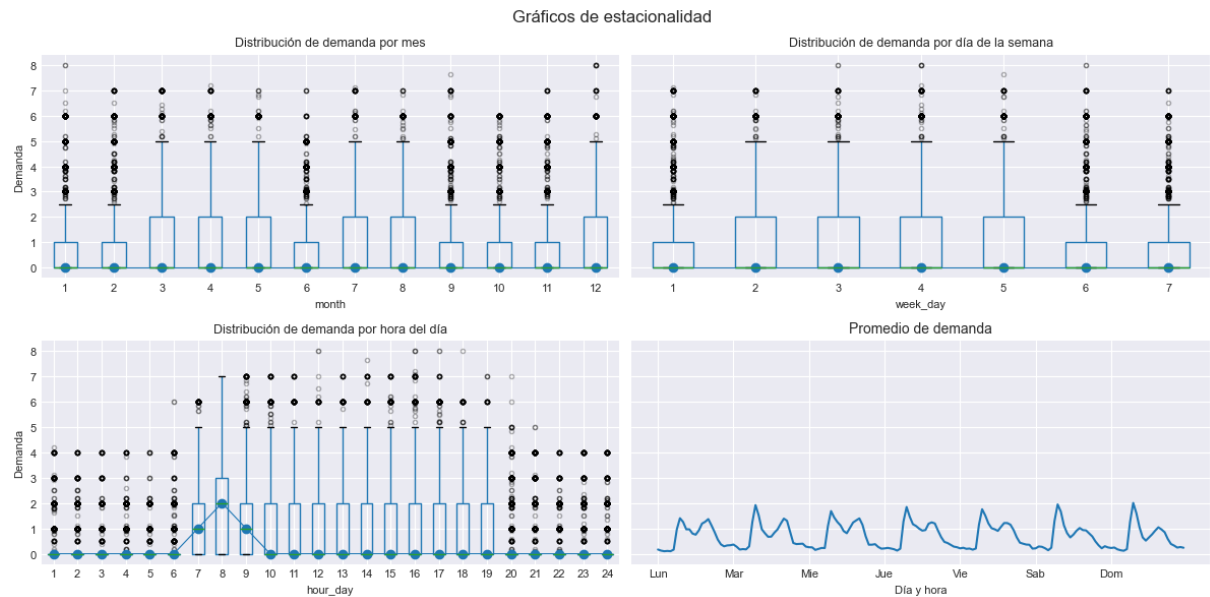


Ilustración 42. Estacionalidad de la serie temporal

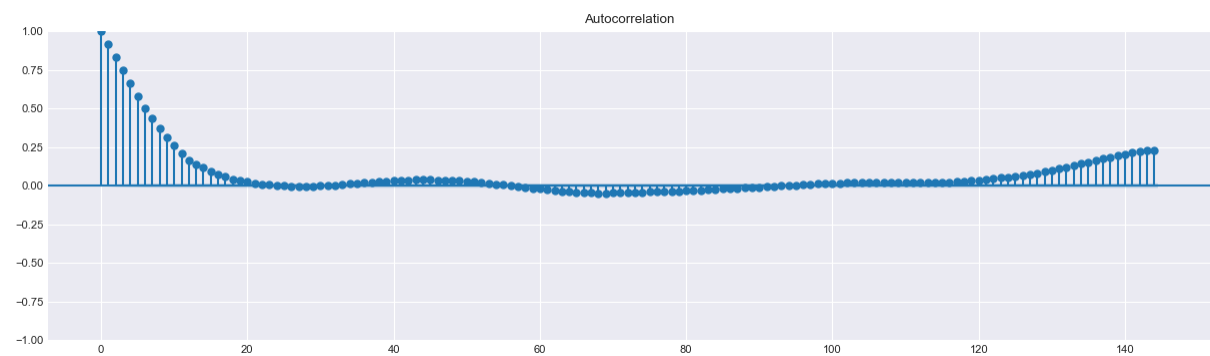


Ilustración 43. Autocorrelación de la variable objetivo para 144 pasos (1 día)

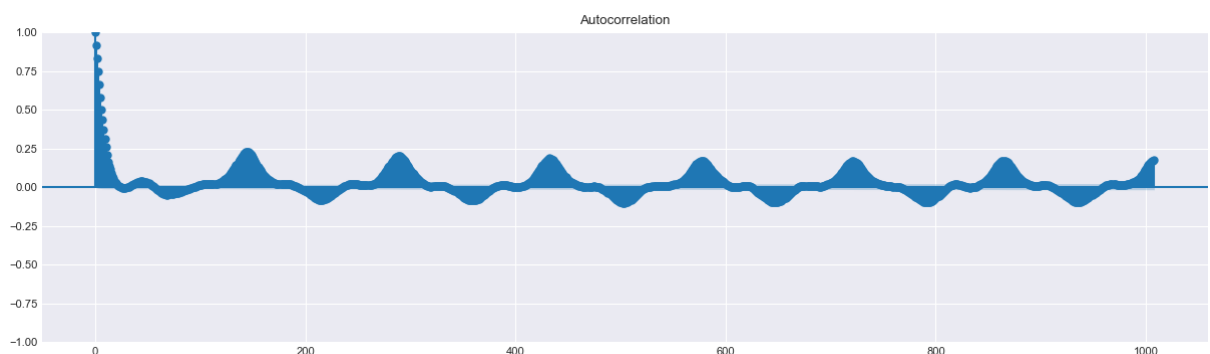


Ilustración 44. Autocorrelación de la variable objetivo para 1008 pasos (1 semana)

3.3.2.3 Estacionariedad

Para la estacionariedad se aplica la prueba Augmented Dickey-Fuller (ADF), arrojo un estadístico de -47.58 y un p-valor de 0.0, lo que permite rechazar la hipótesis nula que

plantea que la serie tiene raíz unitaria (no es estacionaria). Por tanto, se concluye que **la serie es estacionaria**.

La estacionariedad implica que las propiedades estadísticas de la serie, como la media, la varianza y la autocorrelación, se mantienen constantes a lo largo del tiempo; esto es un requisito clave para muchos modelos estadísticos, ya que garantiza que los patrones aprendidos en el pasado sigan siendo válidos en el futuro. Aunque los modelos de *machine learning* como LSTM o XGBoost no requieren que las series temporales sean estacionarias, sigue siendo útil identificar y manejar patrones como la tendencia (cambios sostenidos a lo largo del tiempo) y la estacionalidad (fluctuaciones cíclicas regulares), ya que esto puede mejorar la interpretación y el rendimiento del modelo. En contraste, modelos estadísticos tradicionales como ARIMA sí necesitan que la serie sea estacionaria, por lo que requieren transformaciones previas como la diferenciación para eliminar dichos patrones.

```
ADF Statistic: -47.61345045162687
p-value: 0.0
```

3.3.2.4 Valores atípicos

Para identificar valores atípicos en la serie temporal, se aplicaron dos enfoques complementarios: el **Modified Z-Score** y la **descomposición STL con análisis de residuos**. El primer método, basado en la mediana y la desviación absoluta mediana (MAD), adecuado por su resistencia a la asimetría y a la no normalidad de los datos, esta técnica detectó **59,932 outliers**, lo que representa el **33.38 %** del total de observaciones (Figura 45). Por otro lado, la técnica **STL decomposition**, que separa la serie en componentes de tendencia, estacionalidad y residuo, se identificaron **57,019 valores atípicos** sobre la componente residual mediante criterios como IQR o Modified Z-Score, correspondientes al **31.76 %** de la serie (Figura 46). Estos resultados evidencian una alta proporción de anomalías, y resaltan la importancia de aplicar métodos robustos para su identificación, especialmente en series con estacionalidad marcada.

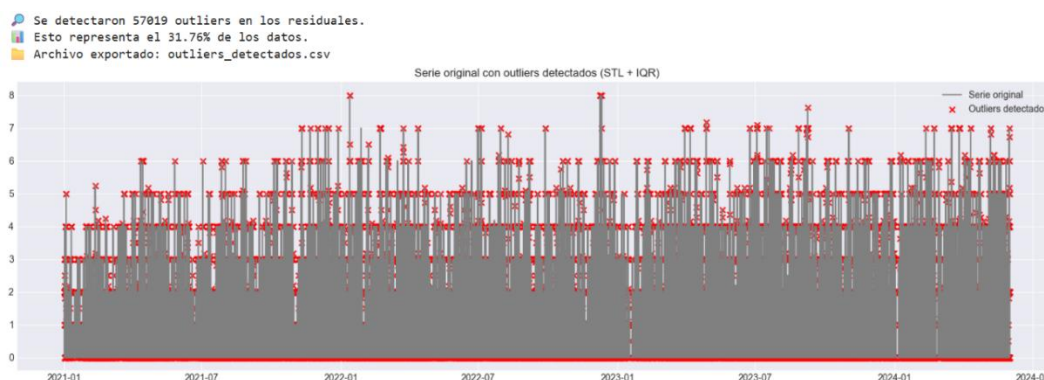


Ilustración 45. Detección de outliers combinando las técnicas STL + IQR

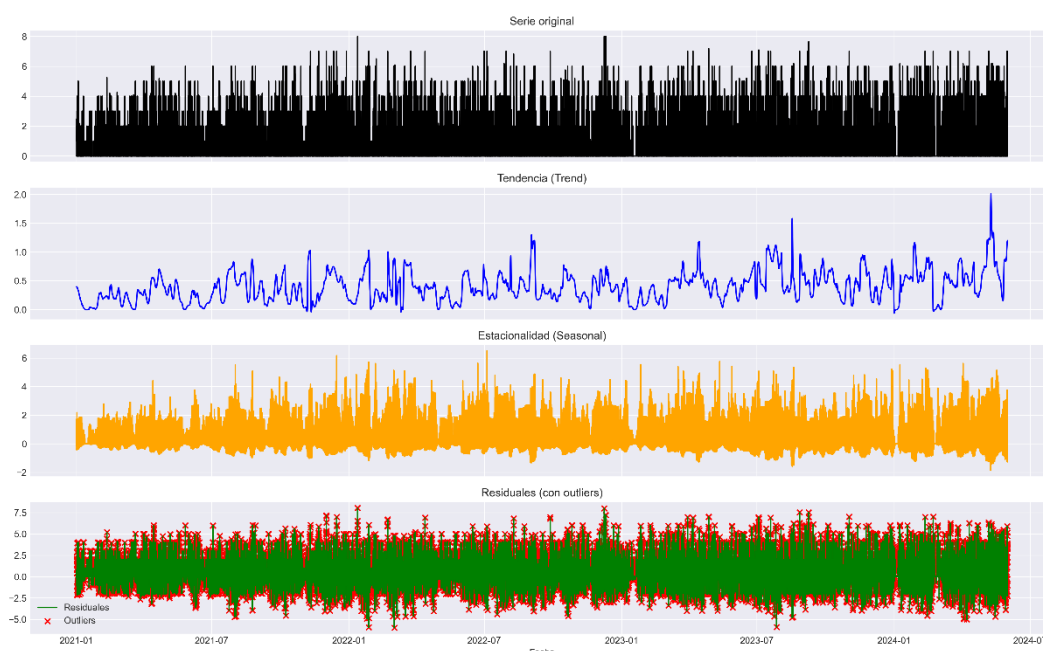


Ilustración 46. Tendencia, estacionalidad y residuos *dataset*

3.3.2.5 Heteroscedasticidad

Para evaluar la presencia de heterocedasticidad condicional en la serie temporal, se aplicó la prueba ARCH (Autoregressive Conditional Heteroskedasticity), el resultado arrojó un p-valor de 0.0000, lo que proporciona evidencia estadísticamente significativa para rechazar la hipótesis nula de homocedasticidad. Esto confirma la presencia de heterocedasticidad condicional, es decir, que la varianza de los errores no se mantiene constante a lo largo del tiempo. Como complemento, la Figura 47 evidencia una marcada variabilidad en la dispersión de los residuos, con periodos de alta volatilidad alternados con fases más estables. Este patrón refuerza la presencia de heterocedasticidad condicional y sugiere que los valores extremos tienden a agruparse en determinados intervalos de tiempo.

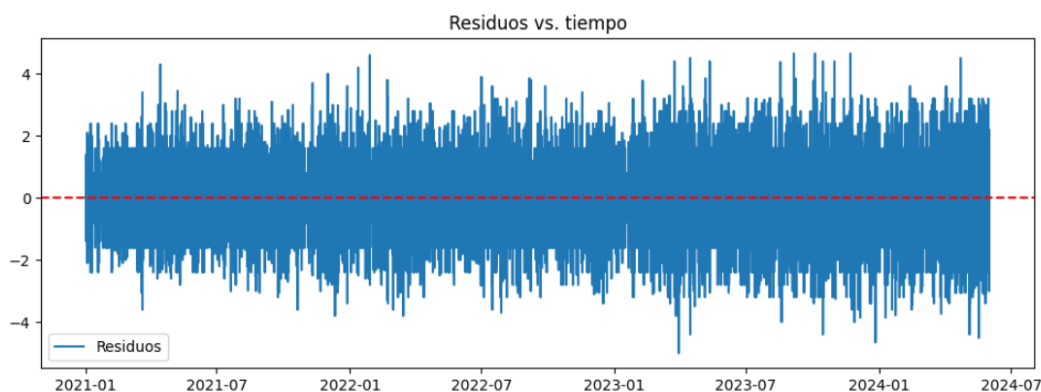


Ilustración 47. Evolución temporal de los residuos calculados a partir de media móvil simple

3.3.2.6 Tendencia

La detección de tendencia en la serie se abordó mediante un análisis estadístico y una inspección visual utilizando suavizado LOESS. Los resultados indican la presencia de una tendencia positiva estadísticamente significativa ($p < 0.0001$). Sin embargo, su efecto práctico es mínimo: el modelo de regresión lineal explica únicamente el 1% de la variabilidad ($R^2 = 0.067$), y el crecimiento estimado es de apenas un 1.1% mensual. Además, se observó una alta autocorrelación en los residuos (Durbin-Watson = 0.45), lo cual sugiere que el modelo no capta adecuadamente la dinámica temporal de los datos. La inspección visual en Figura 48 a través del suavizado LOESS confirma esta conclusión, ya que la curva de tendencia permanece prácticamente estable a lo largo del tiempo. Por tanto, aunque **la serie presenta una tendencia estadística**, esta es tan débil que puede considerarse irrelevante para fines analíticos o predictivos.

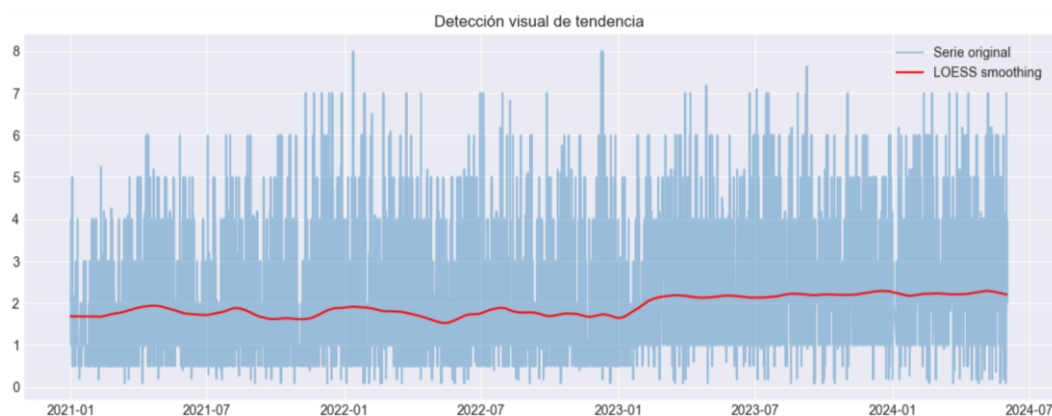


Ilustración 48. Detección visual de la tendencia con LOESS smoothing

3.3.2.7 Intermitencia

La distribución de la variable de remolcadores *-Demand-*, muestra una alta concentración de valores en torno a cero, con picos secundarios en valores discretos como 1, 2 y 3; siendo así esta distribución es fuertemente sesgada hacia la izquierda, lo que indica un patrón *zero-inflated* típico de variables de conteo con presencia frecuente de demanda nula. El análisis de intermitencia confirma este comportamiento: de un total de 179.540 observaciones, 122.479 corresponden a ceros, lo que representa aproximadamente el **68.22% del conjunto de datos**. Esta alta proporción de ceros sugiere largos periodos de inactividad o demanda nula, característicos de contextos con eventos poco frecuentes o alta irregularidad operativa. La presencia de intermitencia no solo afecta la forma de la distribución —alejándola de una distribución normal—, sino que también introduce desafíos relevantes para la modelación predictiva (Figura 49).

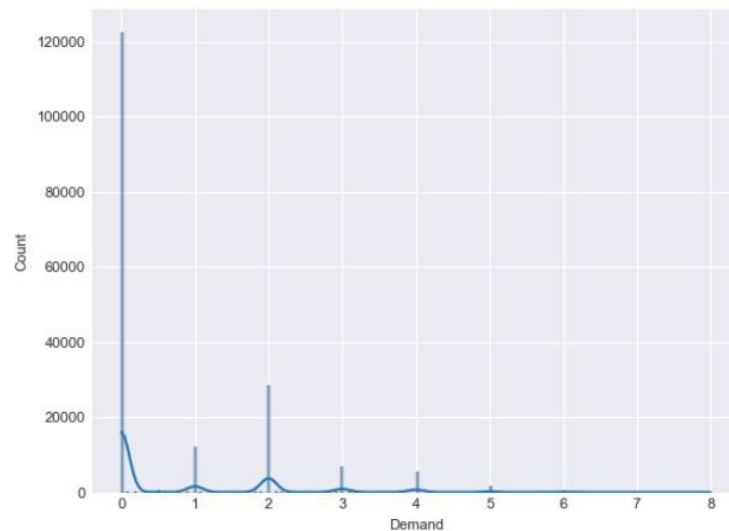


Ilustración 49. Distribución de frecuencia de la variable *Demand*

A continuación, un resumen de las características de la serie temporal:

Propiedad	Valor
Univariada	<input checked="" type="checkbox"/>
Horizonte de pronóstico	Multistep=144
Dominio de valores	[0, 10]
Estacionaria	<input checked="" type="checkbox"/>
Estacionalidad	<input checked="" type="checkbox"/>
Tendencia	Débil
Normalidad	<input checked="" type="checkbox"/> No
Outliers	<input checked="" type="checkbox"/> 31.72%
Intermitencia	<input checked="" type="checkbox"/> 68.24%
Heterocedasticidad	<input checked="" type="checkbox"/>

3.4 FASE 3: Transformación para *forecasting*

3.4.1 Transformación a *dataset* supervisado

Tras completar el preprocesamiento en la fase anterior, se procede a transformar el *dataset* desde una estructura no supervisada hacia un formato supervisado para el modelado de datos con árboles de regresión *gradient boosting*. A continuación, se detalla la explicación de este pseudocódigo de python:

```
from skforecast.recursive import ForecasterRecursive
from sklearn.preprocessing import PowerTransformer
# Crear forecaster univariante recursivo
forecaster = ForecasterRecursive(
    regressor          = regressor_gradient_boosting_seleccionado,
    transformer_y      = PowerTransformer(method='yeo-johnson'),
    transformer_exog= StandardScaler()
    lags               = 144,
    window_features = {
        6: ['mean', 'std', 'min', 'max'],      # 1 hora (6 * 10min)
        12: ['mean', 'std', 'min', 'max'],     # 2 horas
        72: ['mean', 'std', 'min', 'max'],     # 12 horas
        144: ['mean', 'std', 'min', 'max']     # 24 horas
    }
)
```

Para transformar una serie temporal inicialmente no supervisada en un conjunto de datos apto para modelado predictivo, se empleó la librería SKForecast con el módulo `skforecast.recursive`, este proporciona herramientas que convierten modelos de regresión tradicionales (como LightGBM, XGBoost, Random Forest, etc.) en modelos de predicción de series temporales con un orquestador recursivo `ForecasterRecursive`.

Esta técnica recursiva permite estructurar automáticamente los datos en un formato supervisado, donde la variable objetivo a predecir se construye como una versión desplazada temporalmente del propio valor de la serie. Esta configuración del modelo se basó en el uso de 144 rezagos (*lags*), equivalentes a 24 horas de historial a pasos de 10 minutos, lo que permite capturar patrones diarios completos dentro de la ventana de entrenamiento. Junto a los rezagos, se aplicó estadísticas agregadas de ventanas móviles (media, desviación estándar, mínimo y máximo) calculadas sobre distintas escalas temporales (1h, 2h, 12h y 24h) mediante la clase `RollingFeatures`. Estas características agregadas permiten al modelo capturar información contextual y de tendencia en diferentes escalas temporales. La combinación de estas estadísticas con

los rezagos directos mejora la capacidad del modelo para aprender relaciones no lineales y dependencias temporales complejas.

Para reducir la asimetría en la distribución (sin normalización) de la variable objetivo y estabilizar la varianza (heterocedasticidad) de la serie temporal, se aplicó un preprocesamiento con `PowerTransformer` -en su variante *Yeo-Johnson*, *tolerante a ceros*- de `sklearn.preprocessing`. Esta técnica hace que los datos se asemejen más a una distribución normal (gaussiana), facilitando el entrenamiento de modelos sensibles a la escala o la distribución de los datos. Cabe destacar que este módulo facilita el trabajo, invisibilizando la transformación de la variable objetivo 'y' al momento de entrenar y deshaciendo dicha transformación automáticamente al predecir. Esto significa que `PowerTransformer` no solo estandariza, sino que también transforma la distribución de los datos, a diferencia de técnicas convencionales como `StandardScaler`, que únicamente realizan una estandarización sin modificar la forma de la distribución.

En resumen, estas estrategias permitieron transformar la serie temporal en un conjunto supervisado con predictores sólidos basados en rezagos y estadísticas móviles. Además, se aplicó una transformación a la variable objetivo para estabilizar su distribución, paso fundamental para mejorar el rendimiento de modelos de aprendizaje automático como *Gradient Boosting* en tareas de pronóstico temporal.

3.4.2 Ingeniería de características (*time series feature engineering*)

En la transformación, luego de la conversión a un *dataset* supervisado sigue la etapa crucial, con un conjunto de técnicas llamada ingeniería de características en series temporales. Estas transformaciones estadísticas y temporales **mejoran la capacidad predictiva del modelo**, sin ellas, los modelos de *machine learning* no pueden aprovechar la estructura temporal inherente a los datos. A continuación, las características agregadas al *dataset* de la serie temporal:

- Variables temporales: basadas en días festivos (booleano) además de pre-festivo y post-festivo; basadas en el calendario (discreto) con el mes, semana, día de la semana, y hora; basadas en la luz solar (booleano) además de la cantidad de horas (discreto) de salida del sol, puesta de sol y horas de luz solar
- Clima histórico: En la librería *Meteostat* se encuentran disponibles diversas variables climáticas con frecuencia horaria; para este análisis se seleccionó un punto representativo de La Libertad, en la provincia de Santa Elena (Ecuador),

con coordenadas convertidas a formato decimal (-2.1000, -80.9333). Debido a la capacidad de cómputo disponible, se escogieron únicamente las variables principales: temperatura, precipitación, presión atmosférica promedio y velocidad del viento promedio, y se aplicó un remuestreo a intervalos de 10 minutos mediante `resample('10min').mean()` para adecuar la resolución temporal a los requerimientos del estudio.

- **Ventanas móviles:** Se aplica a las variables climáticas, en este caso 1, 6 y 12 horas, calculando la media, máximo y mínimo dentro de cada ventana móvil.
- **Codificación cíclica:** Como su nombre lo expresa, se aplica a las variables cíclicas de tipo numérico discreto, en este caso, a la variable mes (1-12), semana (0-52), día de la semana (0-6), hora (0-23), horas de salida de sol (0-23) y horas de puesta del sol (0-23).

Luego de esta ingeniería de características, a partir de la serie temporal se tiene un nuevo *dataset*, al que se le denomina variables exógenas (Figura 50).

179341 rows × 63 columns

exog.columns

```
Index(['month', 'week', 'day_of_week', 'hour', 'sunrise_hour', 'sunset_hour',
      'daylight_hours', 'is_daylight', 'Holiday', 'holiday_previous_day',
      'holiday_next_day', 'temp', 'temp_window_1H_mean', 'temp_window_1H_max',
      'temp_window_1H_min', 'temp_window_6H_mean', 'temp_window_6H_max',
      'temp_window_6H_min', 'temp_window_12H_mean', 'temp_window_12H_max',
      'temp_window_12H_min', 'prcp', 'prcp_window_1H_mean',
      'prcp_window_1H_max', 'prcp_window_1H_min', 'prcp_window_6H_mean',
      'prcp_window_6H_max', 'prcp_window_6H_min', 'prcp_window_12H_mean',
      'prcp_window_12H_max', 'prcp_window_12H_min', 'wspd',
      'wspd_window_1H_mean', 'wspd_window_1H_max', 'wspd_window_1H_min',
      'wspd_window_6H_mean', 'wspd_window_6H_max', 'wspd_window_6H_min',
      'wspd_window_12H_mean', 'wspd_window_12H_max', 'wspd_window_12H_min',
      'pres', 'pres_window_1H_mean', 'pres_window_1H_max',
      'pres_window_1H_min', 'pres_window_6H_mean', 'pres_window_6H_max',
      'pres_window_6H_min', 'pres_window_12H_mean', 'pres_window_12H_max',
      'pres_window_12H_min', 'month_sin', 'month_cos', 'week_sin', 'week_cos',
      'day_of_week_sin', 'day_of_week_cos', 'hour_sin', 'hour_cos',
      'sunrise_hour_sin', 'sunrise_hour_cos', 'sunset_hour_sin',
      'sunset_hour_cos'],
      dtype='object')
```

Ilustración 50. Variables exógenas por medio de técnicas *feature engineering*

3.4.2.1 Selección de variables exógenas: Correlación de Pearson + Permutation Importance

Se construye la función `feature_analysis` para identificar la relevancia de variables predictoras (exógenas) frente a una variable objetivo, combinando dos técnicas: la correlación de Pearson para medir relaciones lineales, y la Permutation Importance con

un umbral > 0.005 cuantifica el impacto real de cada variable en el rendimiento de un modelo de predicción basado en árboles (Figura 51).

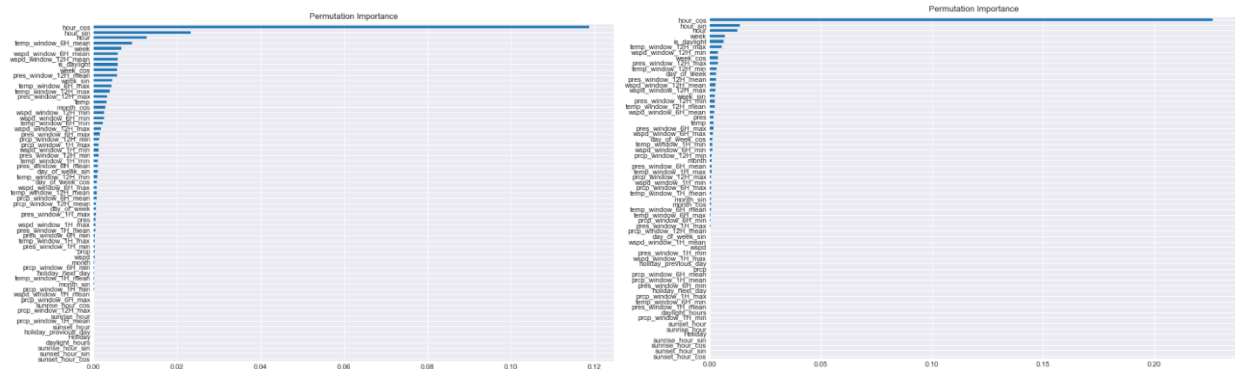
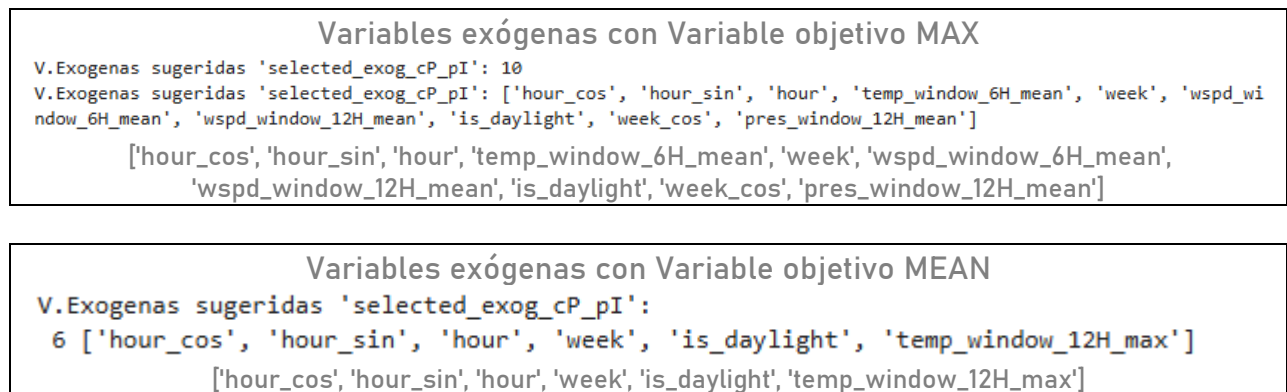


Ilustración 51. Correlación de Pearson & Permutation Importance

3.4.2.2 Selección de variables exógenas: RFECV (Recursive Feature Elimination with Cross-Validation)

En RFECV se empleó como estimador auxiliar el algoritmo LGBMRegressor, esto permite identificar el subconjunto óptimo de predictores que maximiza el rendimiento del modelo en validación cruzada, sin realizar ajustes sobre los hiperparámetros del estimador ni generar nuevas variables. Consiste en evaluar de forma iterativa la importancia de los lags, endógenos, ventanas móviles y variables exógenas preexistentes -en nuestro caso-, eliminando aquellas con menor aporte predictivo de acuerdo a la métrica de error definida (por defecto, MSE); permitiendo reducir la dimensionalidad de las variables y evitar el sobreajuste del modelo (Figura 52).

```
from sklearn.feature_selection import RFECV
from lightgbm import LGBMRegressor

# Paso 1: Definir el selector con LGBM
selector = RFECV(
    estimator=LGBMRegressor(),
    scoring=scorer_s_mse, # por default es 'r2' (coeficiente de determinación)
```



```

step=1,
cv=5, # Buen equilibrio entre tiempo y robustez.
n_jobs=-1,
importance_getter='auto',
min_features_to_select = 20 # Aquí defines el mínimo que debe conservar
)

# Paso 2: Seleccionar características en univariado
selected_lags, selected_window_features, selected_exog = select_features(
    forecaster      = forecaster_custom,
    selector        = selector,
    y               = data_train_val['Demand'],
    exog            = exog_train_val,
    select_only     = None,
    force_inclusion = None,
    subsample       = 0.99, # Coste computacional en la selección de predictores
    random_state    = 123,
    verbose         = True,
)

Recursive feature elimination (RFECV)
-----
Total number of records available: 156504
Total number of records used for feature selection: 154938
Number of features available: 73
  Lags          (n=6)
  Window features (n=4)
  Exog          (n=63)
Number of features selected: 47
  Lags          (n=6) : [3, 12, 18, 144, 288, 1008]
  Window features (n=3) : ['roll_mean_6', 'roll_std_12', 'roll_max_144']
  Exog          (n=38) : ['month', 'week', 'day_of_week', 'hour', 'temp', 'temp_window_1H_mean', 'temp_window_1H_min', 'temp_window_6H_mean', 'temp_window_6H_min', 'temp_window_12H_mean', 'temp_window_12H_min', 'prcp_window_12H_mean', 'wspd', 'wspd_window_1H_mean', 'wspd_window_1H_max', 'wspd_window_1H_min', 'wspd_window_6H_mean', 'wspd_window_6H_max', 'wspd_window_6H_min', 'wspd_window_12H_mean', 'wspd_window_12H_max', 'wspd_window_12H_min', 'pres', 'pres_window_1H_mean', 'pres_window_1H_max', 'pres_window_1H_min', 'pres_window_6H_mean', 'pres_window_6H_max', 'pres_window_6H_min', 'pres_window_12H_mean', 'pres_window_12H_max', 'pres_window_12H_min', 'week_sin', 'week_cos', 'hour_sin', 'hour_cos']
  Tiempo <Univ.FR RFECV+Select: Select_features>: 0:9:27
  Memoria final: 1.66 GB
  Memoria usada: 0.050000000000000044 GB

```

Ilustración 52. El resultado RFECV en variables exógenas

3.5 FASE 4: Modelado y Evaluación

Para la evaluación del modelo, se empleó una estrategia que permite simular escenarios realistas: el backtesting, en conjunto con la clase `TimeSeriesFold` de la librería `Skforecast`. Se optó por aplicar backtesting sin reentrenamiento debido a su bajo costo computacional y a su capacidad para evaluar el desempeño del modelo en múltiples ventanas temporales sin necesidad de ajustar nuevamente los parámetros en cada iteración. `TimeSeriesFold` gestiona la validación cruzada específica para series temporales, dividiendo los datos en bloques de entrenamiento y validación que respetan el orden cronológico, lo que evita el riesgo de *data leakage*. Esta combinación resulta especialmente adecuada para conjuntos de datos grandes, permitiendo obtener

estimaciones realistas del error de predicción con un uso eficiente de los recursos computacionales.

Visual	Modelo	sMSE	RMdSE	Time	MemoriaGB	Predicciones por segundo	Regressor	Enfoque
	UniV_FR Baseline: Backtesting	4.366761	0.007477	00:00:09	2.07	2457.38125	XGBRegressor	max
	UniV_FR Tunning1: Backtesting	4.2496	0.007697	00:00:17	1.63	1277.457938	LGBMRegressor	max
	UniV_FR Tunning1: Backtesting	4.359934	0.008484	00:00:03	0.92	6731.171497	XGBRegressor	max
	UniV_FR Tunning2: Backtesting2 [s_exog_cP_pl]	4.36423	0.008673	00:00:02	1.47	9457.480497	XGBRegressor	max
	UniV_FR Baseline: Backtesting	4.241652	0.009443	00:00:27	1.64	815.15161	LGBMRegressor	max
	UniV_FR Tunning2: Backtesting1 [s_exog_UniV]	4.360981	0.009593	00:00:02	1.37	8950.81024	XGBRegressor	max
	UniV_FR C.Pear-P.Impo: Backtesting2 [s_exog_cP...	4.242359	0.010136	00:00:18	1.75	1206.150638	LGBMRegressor	max
	UniV_FR Tunning2: Backtesting2 [s_exog_cP_pl]	4.238858	0.010359	00:00:18	1.62	1233.237413	LGBMRegressor	max
	UniV_FR RFECV+Select: Backtesting1 [s_exog_UniV]	4.269163	0.010596	00:00:19	1.6	1149.623864	LGBMRegressor	max
	UniV_FR C.Pear-P.Impo: Backtesting2 [s_exog_cP...	4.354174	0.010852	00:00:04	1.3	5629.791128	XGBRegressor	max
	UniV_FR RFECV+Select: Backtesting1 [s_exog_UniV]	4.3513	0.011654	00:00:06	1.13	3459.228224	XGBRegressor	max
	UniV_FR Tunning1: Backtesting	3.452158	0.016228	00:00:35	2.9	623.916269	XGBRegressor	mean
	UniV_FR C.Pear-P.Impo: Backtesting2 [s_exog_cP...	3.435524	0.017616	00:00:17	1.73	1286.008275	LGBMRegressor	mean
	UniV_FR Tunning2: Backtesting2 [s_exog_cP_pl]	3.440199	0.01771	00:00:17	1.75	1252.071384	LGBMRegressor	mean
	UniV_FR Tunning2: Backtesting1 [s_exog_UniV]	3.442185	0.017936	00:00:19	1.75	1165.896892	LGBMRegressor	mean
	UniV_FR Baseline: Backtesting	3.433633	0.018485	00:00:28	1.77	768.01201	LGBMRegressor	mean
si	UniV_FR Baseline: Backtesting	3.467356	0.018918	00:00:09	2.08	2417.469729	XGBRegressor	mean
	UniV_FR Tunning1: Backtesting	3.461965	0.019158	00:00:17	1.51	1320.791001	LGBMRegressor	mean
	UniV_FR Tunning2: Backtesting1 [s_exog_UniV]	3.463196	0.019752	00:00:02	3.5	9391.250593	XGBRegressor	mean
	UniV_FR RFECV+Select: Backtesting1 [s_exog_UniV]	3.427984	0.019984	00:00:19	1.58	1176.322379	LGBMRegressor	mean
	UniV_FR Tunning2: Backtesting2 [s_exog_cP_pl]	3.461881	0.020488	00:00:02	3.57	10335.66931	XGBRegressor	mean
	UniV_FR Baseline ModeloXSerie: Backtesting	4.15564	0.048106	00:00:13	1.43	0.302534	XGBRegressor	max
si	UniV_FR C.Pear-P.Impo: Backtesting2 [s_exog_cP...	3.268971	0.05134	00:01:01	3.26	356.790492	XGBRegressor	mean
	UniV_FR RFECV+Select: Backtesting1 [s_exog_UniV]	3.269631	0.067011	00:00:52	2.86	417.5409	XGBRegressor	mean
	UniV_FR Tunning2: Backtesting1 [s_exog_UniV]	3.846285	0.110026	00:00:19	1.63	1124.216374	LGBMRegressor	max
si	UniV_FR Baseline ModeloXSerie: Backtesting	2.950642	0.297057	00:00:13	3.68	0.303858	XGBRegressor	mean
	UniV_FR ModeloXSerie: Bayesian_search	3.413371	0.52031	00:17:41	1.67	0.000943	XGBRegressor	max
	UniV_FR ModeloXSerie: Bayesian_search	3.420664	0.52071	00:32:32	1.22	0.000512	LGBMRegressor	max
si	UniV_FR Baseline ModeloXSerie: Backtesting	2.38758	0.737647	00:01:30	1.82	0.04456	LGBMRegressor	mean
	UniV_FR ModeloXSerie: Bayesian_search	2.46133	0.790039	00:26:06	2.02	0.000638	XGBRegressor	mean
	UniV_FR ModeloXSerie: Bayesian_search	2.508744	0.893645	00:32:48	1.59	0.000508	LGBMRegressor	mean
	UniV_FR Baseline ModeloXSerie: Backtesting	3.934877	0.959841	00:01:32	1.65	0.043538	LGBMRegressor	max

Ilustración 53. Top en base la métrica RMdSE

Se presentan tres listados de resultados (Figura 53, 54 y 55) con algunos parámetros importantes, se muestran en orden ascendente acorde a la métrica con que se evaluó cada uno de los modelos.

Con respecto al parámetro enfoque, cabe aclarar que se entrenó y validó el modelo con la variable objetivo promedio como en máximo, punto que se mencionó en la fase 2 de preprocesamiento (Figura 38).

Visual	Modelo	sMSE	RMdSE	RMSE	Time	Memoria GB	Predicciones por segundo	Regressor	Enfoque
si	UnIV_FR Baseline ModeloXSerie: Backtesting	2.38758	0.73765	1.38362	00:01:30	2.03	0.044212	LGBMRegressor	mean
	UnIV_FR ModeloXSerie: Bayesian_search	2.46133	0.79004	1.40483	00:55:38	1.53	0.0003	XGBRegressor	mean
	UnIV_FR ModeloXSerie: Bayesian_search	2.50874	0.89365	1.4183	00:31:42	2.03	0.000526	LGBMRegressor	mean
si	UnIV_FR RFECV+Select: Backtesting1 [s_exog_UniV]	2.7466	0.46636	1.48401	00:00:08	2.19	2907.746794	XGBRegressor	mean
si	UnIV_FR Tunning1: Backtesting	2.77232	0.37407	1.49086	00:00:07	2.08	2970.840953	XGBRegressor	mean
si	UnIV_FR Baseline ModeloXSerie: Backtesting	2.95064	0.29706	1.53814	00:00:39	2.14	0.10384	XGBRegressor	mean
	UnIV_FR Tunning2: Backtesting2 [s_exog_cP_pl]	2.96855	0.29824	1.5428	00:00:12	2.11	1768.98977	XGBRegressor	mean
	UnIV_FR Tunning2: Backtesting2 [s_exog_cP_pl]	2.97881	0.27792	1.54547	00:00:18	1.95	1226.958916	LGBMRegressor	mean
	UnIV_FR Tunning2: Backtesting1 [s_exog_UniV]	2.99937	0.25368	1.55079	00:00:19	1.93	1128.680281	LGBMRegressor	mean
	UnIV_FR C.Pear-P.Impo: Backtesting2 [s_exog_cP...	3.01615	0.31972	1.55512	00:00:09	2.31	2560.15767	XGBRegressor	mean
	UnIV_FR Tunning2: Backtesting1 [s_exog_UniV]	3.03251	0.24317	1.55934	00:00:16	2.29	1367.227774	XGBRegressor	mean
	UnIV_FR RFECV+Select: Backtesting1 [s_exog_UniV]	3.03964	0.27674	1.56117	00:00:19	1.78	1135.530036	LGBMRegressor	mean
	UnIV_FR Tunning1: Backtesting	3.05602	0.24856	1.56537	00:00:17	1.75	1280.442265	LGBMRegressor	mean
	UnIV_FR C.Pear-P.Impo: Backtesting2 [s_exog_cP...	3.06009	0.33087	1.56641	00:00:18	1.93	1189.445908	LGBMRegressor	mean
	UnIV_FR ModeloXSerie: Bayesian_search	3.41337	0.52031	1.04873	00:41:20	1.23	0.000403	XGBRegressor	max
	UnIV_FR ModeloXSerie: Bayesian_search	3.42066	0.52071	1.04985	00:31:15	0.92	0.000533	LGBMRegressor	max
	UnIV_FR Baseline: Backtesting	3.43546	0.01829	1.65971	00:00:52	1.79	416.00833	LGBMRegressor	mean
	UnIV_FR Baseline: Backtesting	3.46736	0.01892	1.66611	00:00:32	1.89	684.430237	XGBRegressor	mean
	UnIV_FR Tunning2: Backtesting2 [s_exog_cP_pl]	3.73891	0.46311	1.0976	00:01:34	3.47	231.590239	XGBRegressor	max
	UnIV_FR Tunning2: Backtesting1 [s_exog_UniV]	3.79239	0.66679	1.10542	00:01:22	3.22	266.093812	XGBRegressor	max
	UnIV_FR RFECV+Select: Backtesting1 [s_exog_UniV]	3.82722	0.42001	1.11049	00:00:19	1.72	1142.187327	LGBMRegressor	max
	UnIV_FR C.Pear-P.Impo: Backtesting2 [s_exog_cP...	3.83053	0.42037	1.11097	00:00:18	1.87	1228.756403	LGBMRegressor	max
	UnIV_FR Tunning2: Backtesting2 [s_exog_cP_pl]	3.83053	0.42037	1.11097	00:00:18	1.72	1216.281002	LGBMRegressor	max
	UnIV_FR Tunning1: Backtesting	3.84105	0.42371	1.11249	00:00:17	1.71	1297.514617	LGBMRegressor	max
	UnIV_FR Baseline ModeloXSerie: Backtesting	3.93488	0.95984	1.126	00:01:26	1.76	0.04627	LGBMRegressor	max
	UnIV_FR Tunning1: Backtesting	4.08392	0.10399	1.14713	00:00:44	2.15	492.246095	XGBRegressor	max
	UnIV_FR C.Pear-P.Impo: Backtesting2 [s_exog_cP...	4.08397	0.10399	1.14713	00:01:22	2.54	265.789065	XGBRegressor	max
	UnIV_FR RFECV+Select: Backtesting1 [s_exog_UniV]	4.08445	0.10385	1.1472	00:01:11	2.21	308.246518	XGBRegressor	max
	UnIV_FR Baseline ModeloXSerie: Backtesting	4.15564	0.04811	1.15716	00:00:43	1.63	0.093094	XGBRegressor	max
	UnIV_FR Tunning2: Backtesting1 [s_exog_UniV]	4.21243	0.01131	1.16503	00:00:20	1.86	1109.84766	LGBMRegressor	max
	UnIV_FR Baseline: Backtesting	4.24192	0.00944	1.16853	00:00:27	1.78	819.037884	LGBMRegressor	max
	UnIV_FR Baseline: Backtesting	4.36676	0.00748	1.18619	00:00:10	1.9	2191.346732	XGBRegressor	max

Ilustración 54. Top en base la métrica sMSE

Se presentan tres listados de resultados (Figura 53, 54 y 55) con algunos parámetros importantes, se muestran en orden ascendente acorde a la métrica con que se evaluó cada uno de los modelos.

Con respecto al parámetro enfoque, cabe aclarar que se entrenó y validó el modelo con la variable objetivo promedio como en máximo, punto que se mencionó en la fase 2 de preprocesamiento (Figura 38).

Visual	Modelo	sMSE_R MdSE	RMdSE	sMSE	Time	Memoria aGB	Predicciones por segundo	Regresor	Enfoque
si	UniV_FR Baseline ModeloXSerie: Backtesting	1.56261	0.73765	2.38758	00:01:30	1.2	0.044475	LGBMRegressor	mean
si	UniV_FR Tunning2: Backtesting2 [s_exog_cP_pl]	1.57421	0.27711	2.87131	00:00:02	2.02	10106.38847	XGBRegressor	mean
	UniV_FR ModeloXSerie: Bayesian_search	1.61481	0.62519	2.60444	00:31:24	1.21	0.000531	LGBMRegressor	mean
	UniV_FR ModeloXSerie: Bayesian_search	1.61544	0.62625	2.60463	00:25:56	1.02	0.000643	XGBRegressor	mean
si	UniV_FR Baseline ModeloXSerie: Backtesting	1.62385	0.29706	2.95064	00:00:20	1.38	0.202842	XGBRegressor	mean
	UniV_FR Tunning2: Backtesting1 [s_exog_UniV]	1.63784	0.24317	3.03251	00:00:05	2.37	4034.050605	XGBRegressor	mean
	UniV_FR Tunning1: Backtesting	1.65262	0.25434	3.0509	00:00:17	1.75	1279.00924	LGBMRegressor	mean
	UniV_FR Tunning2: Backtesting1 [s_exog_UniV]	1.6614	0.27961	3.04318	00:00:19	1.13	1141.422333	LGBMRegressor	mean
	UniV_FR RFECV+Select: Backtesting1 [s_exog_UniV]	1.66227	0.28142	3.04313	00:00:19	1.65	1146.565271	LGBMRegressor	mean
	UniV_FR Tunning2: Backtesting2 [s_exog_cP_pl]	1.664	0.29613	3.03187	00:00:17	1.12	1263.7746	LGBMRegressor	mean
	UniV_FR C.Pear-P.Impo: Backtesting2 [s_exog_cP...	1.68644	0.31475	3.05812	00:00:18	1.77	1238.129574	LGBMRegressor	mean
si	UniV_FR Baseline: Backtesting	1.72689	0.0183	3.43547	00:00:28	1.83	767.296716	LGBMRegressor	mean
	UniV_FR Baseline: Backtesting	1.74314	0.01892	3.46736	00:00:09	2.03	2384.802076	XGBRegressor	mean
si	UniV_FR RFECV+Select: Backtesting1 [s_exog_UniV]	1.76113	0.58328	2.93897	00:00:26	2.1	852.006619	XGBRegressor	mean
	UniV_FR Tunning1: Backtesting	1.76605	0.65571	2.87639	00:00:19	1.94	1177.996454	XGBRegressor	mean
	UniV_FR C.Pear-P.Impo: Backtesting2 [s_exog_cP...	1.79708	0.73982	2.85433	00:00:23	2.23	950.724334	XGBRegressor	mean
	UniV_FR ModeloXSerie: Bayesian_search	1.94723	0.42485	3.46961	00:26:19	1.13	0.000633	XGBRegressor	max
	UniV_FR ModeloXSerie: Bayesian_search	1.9708	0.5212	3.42041	00:36:58	0.71	0.000451	LGBMRegressor	max
	UniV_FR Tunning2: Backtesting2 [s_exog_cP_pl]	2.00941	0.25318	3.76563	00:00:18	1.42	1234.159739	LGBMRegressor	max
	UniV_FR Tunning1: Backtesting	2.02904	0.03049	4.02759	00:00:11	1.16	2048.283332	XGBRegressor	max
	UniV_FR Tunning2: Backtesting2 [s_exog_cP_pl]	2.05804	0.40948	3.7066	00:00:22	1.48	992.61141	XGBRegressor	max
	UniV_FR C.Pear-P.Impo: Backtesting2 [s_exog_cP...	2.05804	0.40948	3.7066	00:00:22	1.09	988.345064	XGBRegressor	max
	UniV_FR Tunning2: Backtesting1 [s_exog_UniV]	2.0594	0.41059	3.70821	00:00:22	1.65	995.232459	XGBRegressor	max
	UniV_FR RFECV+Select: Backtesting1 [s_exog_UniV]	2.08773	0.40984	3.76563	00:00:23	0.85	945.513462	XGBRegressor	max
	UniV_FR Baseline ModeloXSerie: Backtesting	2.10187	0.04811	4.15564	00:00:13	1.68	0.313876	XGBRegressor	max
	UniV_FR Tunning2: Backtesting1 [s_exog_UniV]	2.11413	0.01125	4.21701	00:00:19	1.42	1146.269308	LGBMRegressor	max
	UniV_FR RFECV+Select: Backtesting1 [s_exog_UniV]	2.12361	0.42001	3.82722	00:00:19	1.27	1131.210894	LGBMRegressor	max
	UniV_FR C.Pear-P.Impo: Backtesting2 [s_exog_cP...	2.12545	0.42037	3.83053	00:00:18	1.43	1211.625785	LGBMRegressor	max
	UniV_FR Baseline: Backtesting	2.12568	0.00944	4.24192	00:00:27	1.64	818.254601	LGBMRegressor	max
	UniV_FR Tunning1: Backtesting	2.13238	0.42371	3.84105	00:00:17	1.46	1293.995213	LGBMRegressor	max
	UniV_FR Baseline: Backtesting	2.18712	0.00748	4.36676	00:00:09	1.61	2396.655741	XGBRegressor	max
	UniV_FR Baseline ModeloXSerie: Backtesting	2.44736	0.95984	3.93488	00:01:25	1.36	0.046788	LGBMRegressor	max

Ilustración 55. Top en base la métrica sMSE+RMdSE

Para identificar los tres mejores modelos según la métrica, se aplicó un enfoque heurístico basado en una evaluación visual de los diez primeros modelos del listado. Esta evaluación consistió en comparar gráficamente la variable respuesta con su predicción. En muchos casos, a pesar de que algunos modelos mostraban un buen desempeño según la métrica, su comportamiento visual no era consistente con ese resultado. Por ejemplo, en la Figura 53 se muestra el primer modelo del listado con un RMdSE=0.007477, su gráfico correspondiente a la Figura 56, aquí puede notarse que el modelo XGBoost tiene mayor precisión en los rangos bajos de la serie, pero no logra capturar adecuadamente los eventos extremos o picos, por lo que limita su capacidad predictiva.

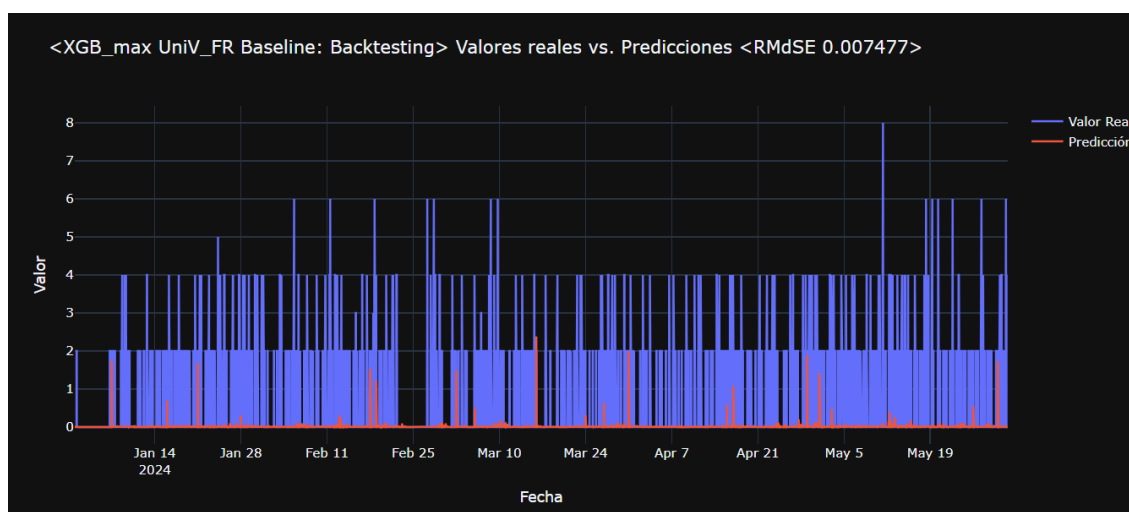


Ilustración 56. Primer modelo de la métrica RMdSE

En base a este enfoque heurístico, en la Figura 57 se detallan los tres mejores modelos por métrica. Esta selección representa las opciones más destacadas en términos de desempeño del pronóstico. Luego, a estos modelos seleccionados se aplica Forecasting Probabilístico, con el fin de escoger aquel modelo que proporcione mayor validez y confiabilidad, expresado términos de porcentaje de probabilidad y eficiencia.

Como parte del análisis comparativo entre métodos de estimación de intervalos de predicción, se realizaron dos procedimientos complementarios:

1. En primer lugar, se calculó una métrica denominada **eficiencia relativa**, definida como la relación entre la cobertura empírica (en porcentaje) y el ancho medio del intervalo. Esta métrica permite cuantificar qué tan eficaz es un modelo para cubrir valores reales con el menor ancho posible, es decir, qué tanto "cubre más con menos". Un valor más alto en esta métrica indica un mejor balance: mayor cobertura utilizando intervalos más estrechos (Ecuación 4 y Figura 57).
2. En segundo lugar, se construyó visualizaciones gráficas que permiten comparar la cobertura empírica y el ancho medio de los intervalos generados por cada modelo y método probabilístico (calibration, residuos in-sample y out-sample). Por lo que se identificó patrones y comportamientos en relación con su capacidad para balancear precisión y confiabilidad (Figura 58).

Ecuación 3

$$\text{Eficiencia} = \frac{\text{Cobertura (\%)}}{\text{Ancho del intervalo}}$$

sMSE_RmdSE														
TOP	ID	Modelo	sMSE_RmdSE	sMSE	RmdSE	Area_Residuos_inSample_80%	Area_Residuos_outSample_80%	Area_Calibration_80%	Cobertura_Residuos_inSample_80%	Cobertura_Residuos_outSample_80%	Cobertura_Calibration_80%	Efficiencia_Residuos_inSample_80%	Efficiencia_Residuos_outSample_80%	Efficiencia_Calibration_80%
0	sMSE_RmdSE_1LGBM	Univ_FR Baseline ModeloXSeries: Backtesting	1.562613	2.38758	0.737647	63592.5	136876.71	68480.37	46.53	85.11	77.15	0.000732	0.000622	0.001127
1	sMSE_RmdSE_2XGB	Univ_FR Tunning2: Backtesting2 [s_exog_cp_pi]	1.574208	2.871307	0.277109	18682.55	20416.48	65462.94	16.58	66.44	78.49	0.000887	0.003254	0.001199
2	sMSE_RmdSE_3XGB	Univ_FR Baseline ModeloXSeries: Backtesting	1.623849	1.623849	1.623849	68342.75	126741.53	70903.76	64.41	69.14	76.02	0.000942	0.000546	0.001072
3	sMSE_RmdSE_4LGBM	Univ_FR Baseline: Backtesting	1.726887	3.435473	0.018302	24453.14	67654.87	64314.63	53.83	44.00	79.26	0.002201	0.00065	0.001232
4	sMSE_RmdSE_5XGB	Univ_FR RFECV>Select: Backtesting1 [s_exog_Univ]	1.761129	2.938974	0.583284	19754.18	19218.6	76392.27	8.61	66.38	87.82	0.000436	0.003454	0.00115

sMSE														
TOP	ID	Modelo	sMSE_RmdSE	sMSE	RmdSE	Area_Residuos_inSample_80%	Area_Residuos_outSample_80%	Area_Calibration_80%	Cobertura_Residuos_inSample_80%	Cobertura_Residuos_outSample_80%	Cobertura_Calibration_80%	Efficiencia_Residuos_inSample_80%	Efficiencia_Residuos_outSample_80%	Efficiencia_Calibration_80%
0	sMSE_1LGBM	Univ_FR Baseline ModeloXSeries: Backtesting	---	2.38758	0.737647	63594.81	136876.71	68481.98	46.53	85.11	77.15	0.000732	0.000622	0.001127
1	sMSE_2XGB	Univ_FR RFECV>Select: Backtesting1 [s_exog_Univ]	---	2.746601	0.466364	19465.28	24496.07	64476.43	7.92	66.88	76.46	0.000407	0.00273	0.001186
2	sMSE_3XGB	Univ_FR Tunning1: Backtesting	---	2.772319	0.374065	22437.85	21000.32	65670.47	6.81	66.40	82.14	0.000304	0.003162	0.001251
3	sMSE_4XGB	Univ_FR Baseline ModeloXSeries: Backtesting	---	2.950642	0.297057	68342.75	126741.53	70903.76	64.41	69.14	76.02	0.000942	0.000546	0.001072

RmdSE														
TOP	ID	Modelo	sMSE_RmdSE	sMSE	RmdSE	Area_Residuos_inSample_80%	Area_Residuos_outSample_80%	Area_Calibration_80%	Cobertura_Residuos_inSample_80%	Cobertura_Residuos_outSample_80%	Cobertura_Calibration_80%	Efficiencia_Residuos_inSample_80%	Efficiencia_Residuos_outSample_80%	Efficiencia_Calibration_80%
0	RmdSE_1XGB	Univ_FR Baseline ModeloXSeries: Backtesting	---	3.467356	0.018918	22150.89	43387.95	71694.87	59.24	61.08	82.93	0.002674	0.001408	0.001157
1	RmdSE_2XGB	Univ_FR C-Pear-P-imp: Backtesting2 [s_exog_cp_pi]	---	3.268971	0.05134	12030.64	61509.52	82448.13	36.05	81.11	81.87	0.002997	0.001319	0.000993
2	RmdSE_3XGB	Univ_FR Baseline ModeloXSeries: Backtesting	---	2.950642	0.297057	68342.75	126741.53	70903.76	64.41	69.14	76.02	0.000942	0.000546	0.001072
3	RmdSE_4LGBM	Univ_FR Baseline ModeloXSeries: Backtesting	---	2.38758	0.737647	63592.48	136876.72	68479.65	46.53	85.11	77.15	0.000732	0.000622	0.001127

Ilustración 57. Forecasting Probabilístico a los modelos seleccionados.

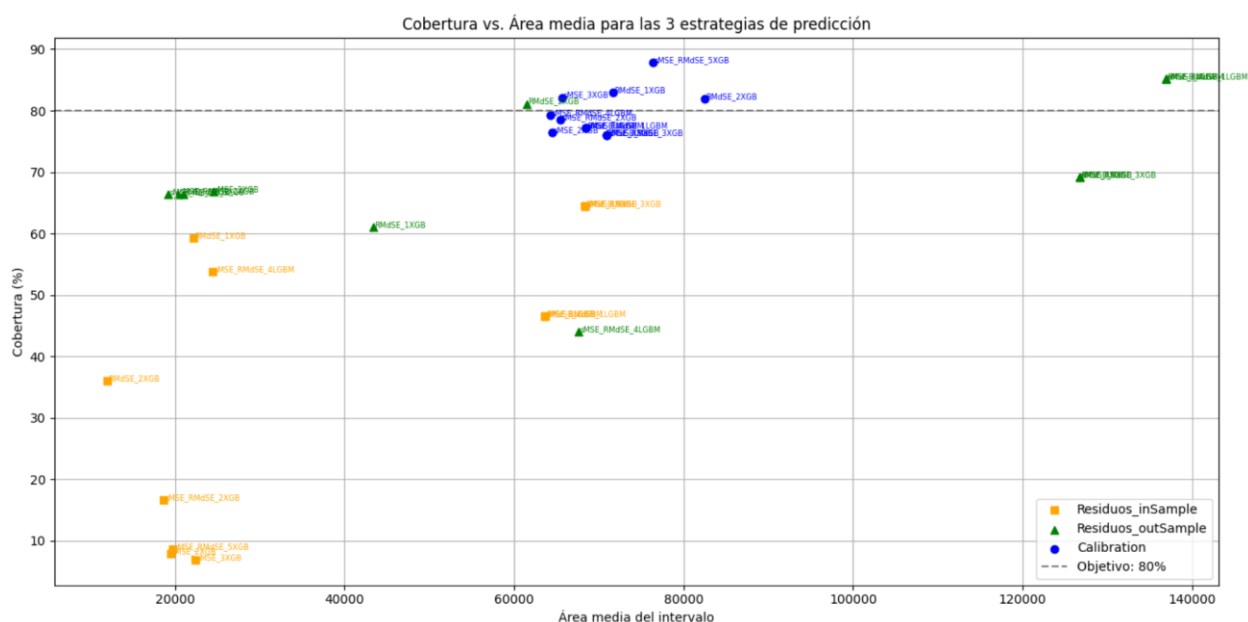


Ilustración 58. Cobertura vs. área para las estrategias de Forecasting Probabilístico

Por el traslape de la Figura 58, se agregó un gráfico (Figura 59) de dispersión para mayor interpretación de la eficiencia entre los tres métodos. Aquí, bootstrap con residuos out-of-sample muestra alta variabilidad, en algunos modelos alcanzan picos de eficiencia muy buenos, pero en otros caen notablemente, a diferencia de calibration/conformal que presentan eficiencias muy estables entre modelos.

Basado en este contexto y la literatura, la estrategia Conformal es el método más robusto y válido, mientras que Bootstrap OoS puede superar en eficiencia a veces, pero no con la misma consistencia; Bootstrap in-sample es el menos fiable, pues tienden a ser optimistas.

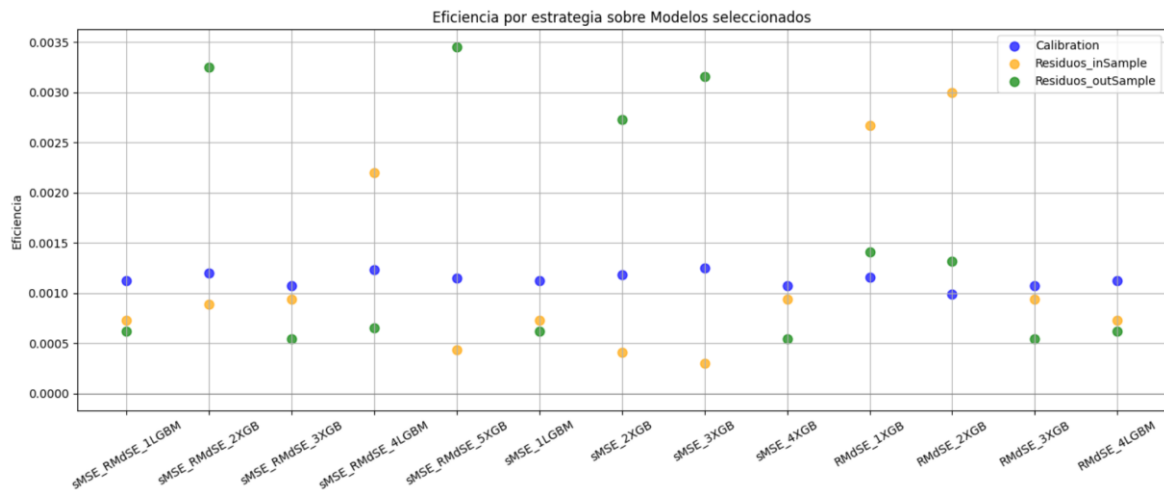


Ilustración 59. Diagrama de dispersión de la eficiencia de métodos probabilístico en modelos seleccionados

Basado en la Figura 59, se agrega la Figura 60 para identificar los métodos probabilísticos forecasting que alcanzaron la cobertura calculada ($\geq 80\%$). Podemos notar que el modelo RMdSE_2XGB no solo destaca por la calibración con 81.11%, sino que además logró mantener la eficiencia en el escenario más exigente (datos no vistos) con 81.87%. Un modelo con buena calibración y residuos out-sample eficientes significa que predice con intervalos confiables en escenarios reales.

El método Calibration corrige los intervalos para garantizar cubrir el valor real → In-sample valida internamente con datos vistos (test sobre train+validation) → Out-of-sample prueba la generalización con datos no vistos (validation sobre train).

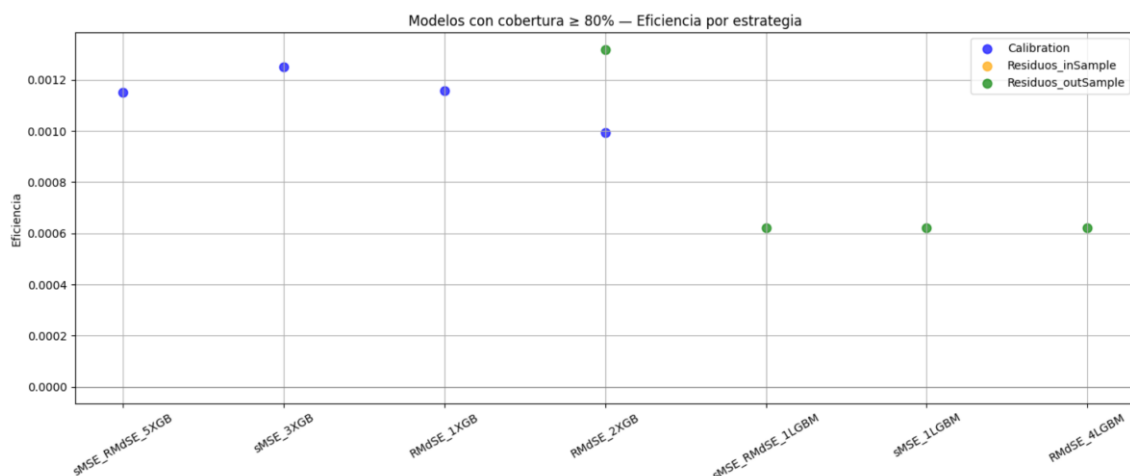


Ilustración 60. Métodos probabilísticos que alcanzaron la cobertura calculada $\geq 80\%$

Continuando con los modelos que alcanzaron la cobertura del 80%, en la Figura 61 se discrimina las diez primeras eficiencias entre cobertura empírica y ancho medio del

intervalo. Aquí, se aprecia que solo 7 modelos cumplen con el requerimiento, también una vez más se evidencia que la mejor eficiencia entre los modelos es RMdSE_2XGB, con el método Bootstrapped residuals out-sample en primer lugar y en quinto lugar con el método ConformalIntervalCalibrator.

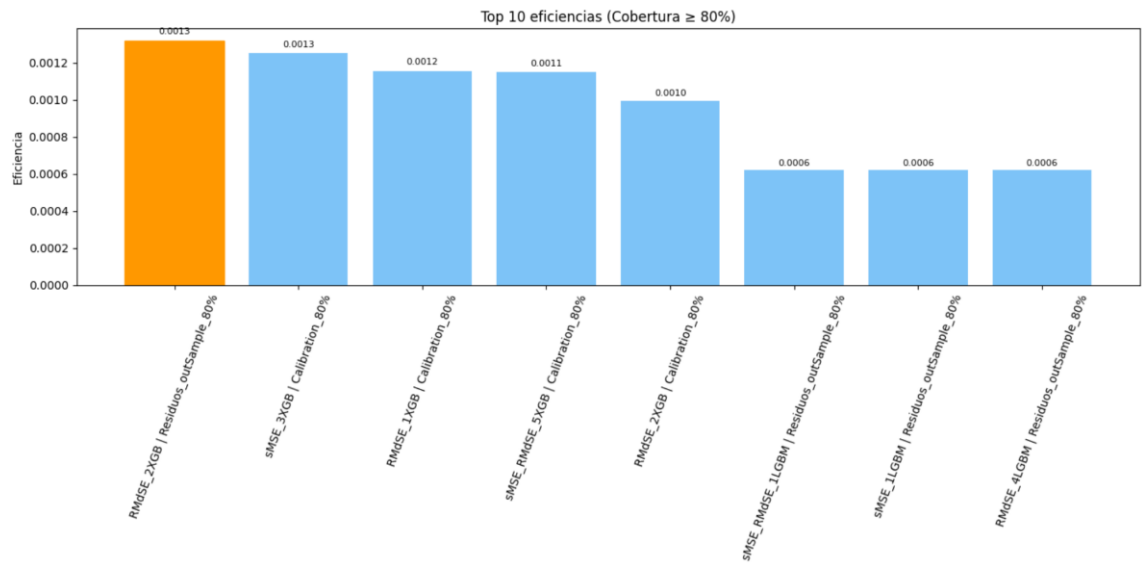


Ilustración 61. Top 10 eficiencias filtrado por Cobertura ≥ 80%

En consonancia con las Figuras 58, 60 y 61 se tiene el top final guardando la mayor precisión (métrica) y confiabilidad (eficiencia) posible de los modelos que alcanzaron la cobertura deseada del 80% de probabilidad de contener el valor real en los pronósticos (Figura 62).

La estrategia evidencia que la elección del método de intervalos no solo debe considerar la cobertura deseada, sino también el ancho promedio generado, especialmente cuando se busca maximizar la confianza sin perder precisión.

sMSE_RMdSE														
TOP		ID	sMSE_RMdSE	sMSE	RMdSE	Area_Residuos_inSample_80%	Area_Residuos_outSample_80%	Area_Calibration_80%	Cobertura_Residuos_inSample_80%	Cobertura_Residuos_outSample_80%	Cobertura_Calibration_80%	Eficiencia_Residuos_inSample_80%	Eficiencia_Residuos_outSample_80%	Eficiencia_Calibration_80%
Top 4	4	sMSE_RMdSE_5XGB	1.761129	2.938974	0.583284	19754.18	19218.6	76392.27	8.61	66.38	87.82	0.000436	0.003454	0.00115

sMSE														
TOP		ID	sMSE_RMdSE	sMSE	RMdSE	Area_Residuos_inSample_80%	Area_Residuos_outSample_80%	Area_Calibration_80%	Cobertura_Residuos_inSample_80%	Cobertura_Residuos_outSample_80%	Cobertura_Calibration_80%	Eficiencia_Residuos_inSample_80%	Eficiencia_Residuos_outSample_80%	Eficiencia_Calibration_80%
Top 2	2	sMSE_3XGB	--	2.772319	0.374065	22437.85	21000.32	65670.47	6.81	66.40	82.14	0.000304	0.003162	0.001251

RMdSE														
TOP		ID	sMSE_RMdSE	sMSE	RMdSE	Area_Residuos_inSample_80%	Area_Residuos_outSample_80%	Area_Calibration_80%	Cobertura_Residuos_inSample_80%	Cobertura_Residuos_outSample_80%	Cobertura_Calibration_80%	Eficiencia_Residuos_inSample_80%	Eficiencia_Residuos_outSample_80%	Eficiencia_Calibration_80%
Top 3	0	RMdSE_1XGB	---	3.467356	0.018918	22150.89	43387.95	71694.87	59.24	61.08	82.93	0.002674	0.001408	0.001157
Top 1	1	RMdSE_2XGB	---	3.268971	0.05134	12030.64	61509.52	82448.13	36.05	81.11	81.87	0.002997	0.001319	0.000993

Ilustración 62. Top Final de los mejores modelos acorde a la precisión y confiabilidad

3.6 FASE 5: Interpretación del conocimiento

En la investigación fue posible identificar los modelos con mejor desempeño relativo dentro de cada enfoque (Figura 62).

Esta última fase de interpretación se basará en el modelo top RMdSE_2XGB llamado *UniV_FR C.Pear-P.Impo: Backtesting2 [s_exog_cP_pl]*. Tal modelo se posicionó como el más eficiente, logrando una cobertura del 81.11% con un ancho medio de 61,509.52 unidades, permitiendo alcanzar la eficiencia relativa más alta entre todos los modelos evaluados con una precisión de $\text{RMdSE}=0.051340$ y $\text{sMSE}=3.268971$.

3.6.1 Precisión del modelo top1: RMdSE_2XGB

Recordemos que la precisión (análoga a la validez) se refiere a la cercanía de los pronósticos puntuales respecto al valor observado y suele evaluarse con métricas de error.

En la Figura 63 se muestra las predicciones con respecto a los valores, la gráfica muestra que el modelo tiende a subestimar los valores reales sin capturar adecuadamente los picos ni la amplitud de las fluctuaciones. Como precisión del modelo RMdSE_2XGB tiene un $\text{RMdSE}=0.051340$, y $\text{sMSE}=3.268971$, $\text{msMAPE}=3.794676$.

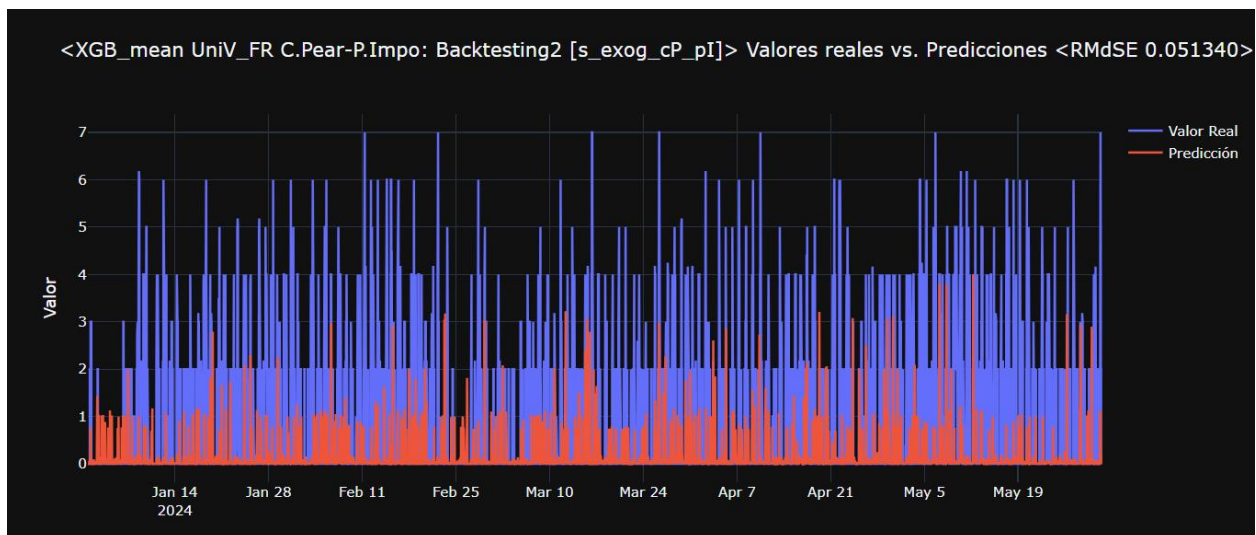


Ilustración 63. Mejor modelo: predicciones vs. valores reales

Sin embargo, es importante señalar que, si bien este modelo despunta en eficiencia, las métricas de precisión utilizadas —Root Median Squared Error (RMdSE) y scaled Mean Squared Error (sMSE)— aún presentan valores alejados del valor real. Dado que ambas métricas penalizan los errores de predicción en relación con la dispersión de los datos,

su valor óptimo debe tender a cero. Siendo así, los valores observados indican que el modelo, aún presentan márgenes de error significativos. Esto sugiere que, si bien se ha logrado una mejora en eficiencia de cobertura, la precisión absoluta de las predicciones continúa siendo un desafío.

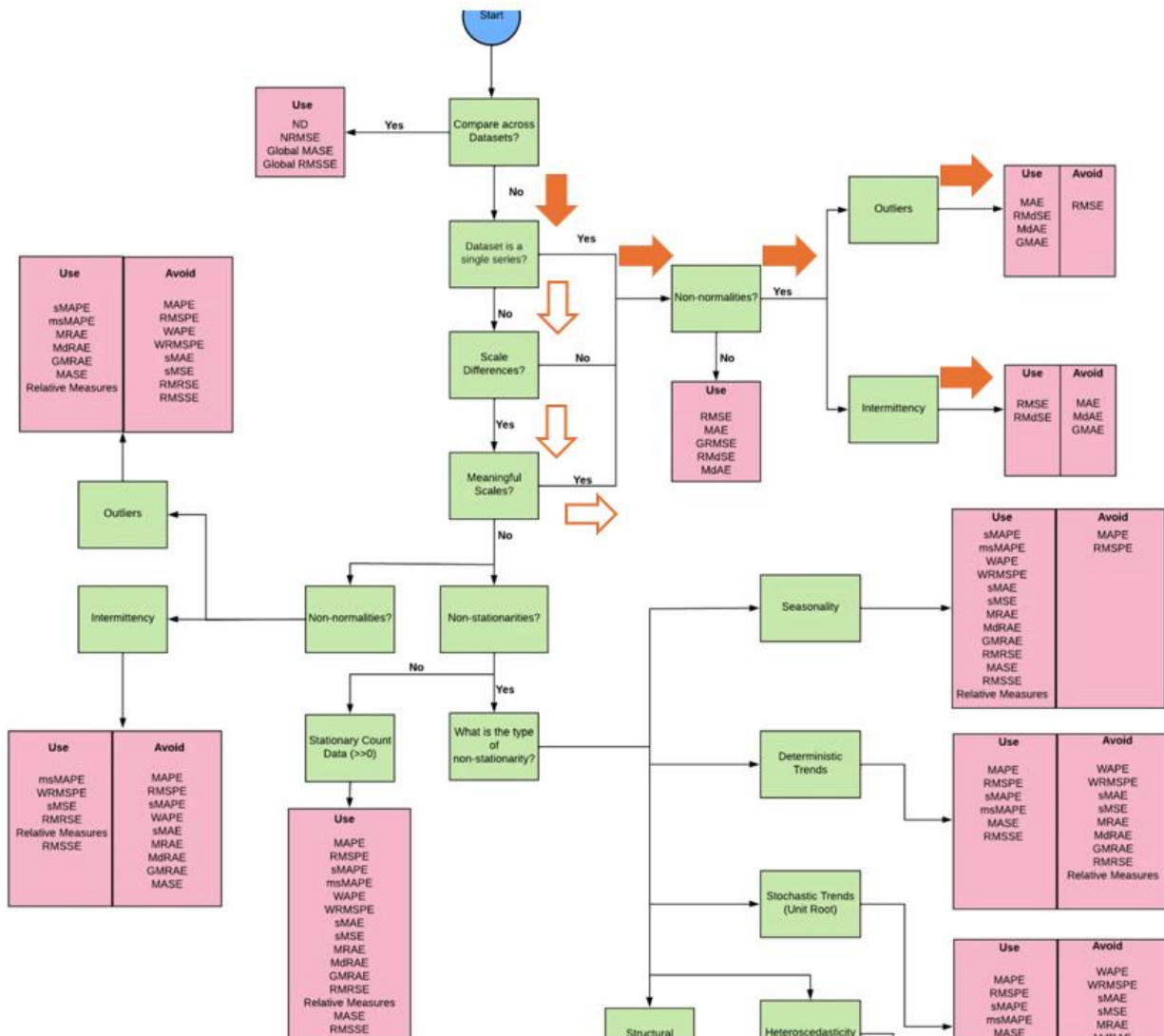


Ilustración 64. Diagrama de flujo para la selección de la medida del error de predicción. **Fuente:** Hewamalage *et al.*, 2023

La razón de RMdSE se sustenta en el flujograma de Hewamalage *et al.* (2023), quienes aconsejan la elección de métricas acorde a las propiedades de la serie. En este estudio, se trabaja con una única serie univariada, por lo que no aplica la comparación entre conjuntos. También, se ha identificado que la serie no sigue una distribución normal, excluyendo métricas sensibles a la media como el RMSE. A esto se suma la presencia significativa de outliers (31.72%), que requiere métricas robustas como la mediana. Paralelamente, la serie presenta una alta intermitencia (68.24%), lo que también descarta métricas basadas en el error absoluto medio, como el MAE (Figura 64).

En el caso de multivariada (flecha contorno) se plantea si existen diferencias de escala dentro de la serie, se responde que sí debido a la heterocedasticidad (variación no constante del error). Luego, la ruta sigue hacia la evaluación de si la escala es significativa (es decir, si tiene un significado práctico), también se responde que sí, ya que los valores de la serie representan demanda real y tienen una escala interpretable.

Ambas condiciones —outliers e intermitencia— convergen en recomendar el uso de la métrica Root Median Squared Error (RMdSE), ya que esta combina la robustez ante valores extremos con estabilidad en contextos donde los datos reales presentan ceros frecuentes (Figura 64).

Desde esta perspectiva cuantitativa, las métricas de desempeño obtenidas para el modelo reflejan una combinación de fortalezas y oportunidades de mejora. El valor de $RMdSE = 0.051340$, al estar expresado en las mismas unidades que la variable objetivo, indica que la mayoría de las predicciones se encuentran cercanas a los valores reales – a cero por la intermitencia con un 68.22%-, lo cual respalda el comportamiento típico de la serie. A la par, el valor de $sMSE = 3.268971$ resulta considerablemente superior a 1.0, lo que implica que el error cuadrático medio del modelo es más de tres veces mayor que el valor medio de la serie, señalando un ajuste general deficiente en términos absolutos. Esta discrepancia sugiere que, aunque el modelo logra capturar bien la mediana de los errores (como muestra el RMdSE), existen errores extremos que elevan el promedio cuadrático. En conjunto, estas métricas sugieren que el modelo ofrece un buen nivel de precisión en términos relativos y mediana de errores, aunque la dispersión de algunos errores grandes está afectando negativamente su ajuste global.

3.6.2 Confiabilidad del modelo top1: RMdSE_2XGB

Recordemos que Forecasting Probabilístico no se trata sólo de predecir bien (precisión), sino de saber cuán seguro se está al predecir (confiabilidad).

En la Figura 65 la cuantificación de la incertidumbre del pronóstico. El uso de intervalos de predicción contruidos con bootstrapping de residuos out-of-sample no condicionados sin imponer supuestos fuertes (e.g. normalidad), arroja una cobertura observada del 81.11% para un objetivo del 80% sugiere una calibración adecuada con un área/ancho del intervalo de 61,509.5. Esta técnica es agnóstica al modelo, sencilla de implementar y muy práctica para decisiones de riesgo (stock, capacidad, alertas).

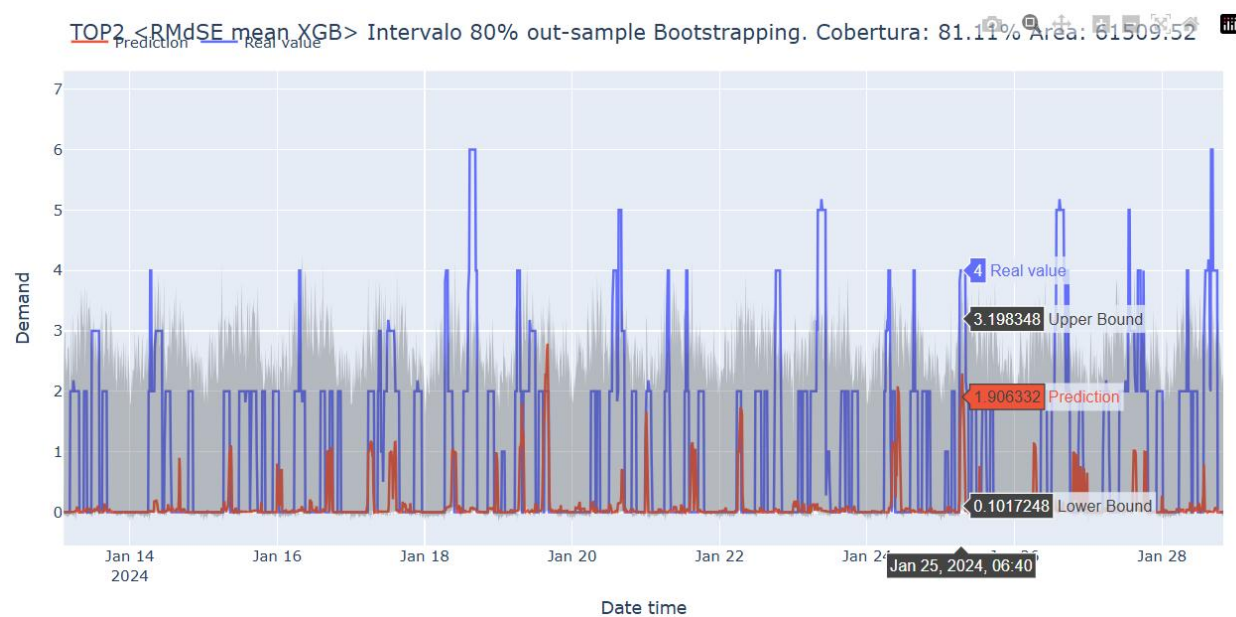
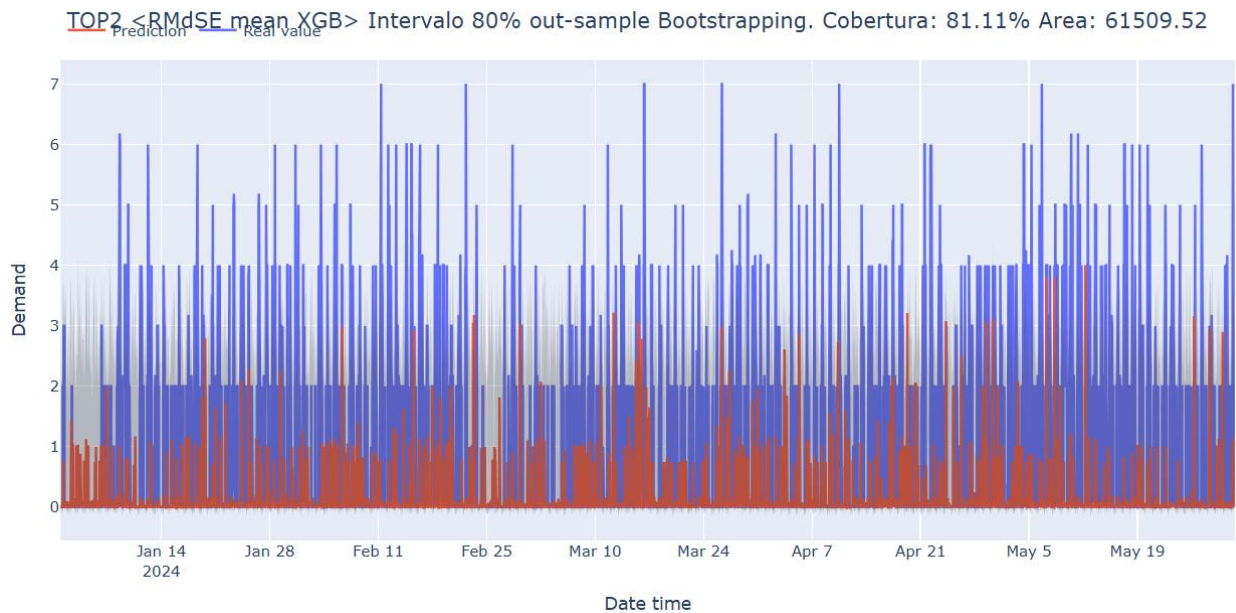


Ilustración 65. Mejor modelo: intervalo de predicción 80% con la técnica Bootstrapping
residuales Out-of-sample (no condicionados)

Los residuos OOS no condicionados, muestran más valores negativos que positivos, si el residuo se define como $y_{true} - y_{pred}$, el modelo tiende a sobreestimar con mayor frecuencia (residuo negativo). Por otro lado, cuando el error es positivo, suele ser más grande la cola derecha es larga (valores hasta ~ 8). Además, la distribución no es gaussiana y presenta rasgos de bimodalidad/heterogeneidad (Figura 66).

El ACF revela autocorrelación que decae recién hacia lag 20–25, por lo que los residuos no son i.i.d. (significa independientes e idénticamente distribuidos) y tienden a agruparse en rachas. Cuando hay dependencia temporal, los errores aparecen en rachas (autocorrelación). Por ejemplo, si a las 10:00 el modelo se queda corto en +4 (real 9,

predicho 5), es probable que a las 10:10, 10:20 y 10:30 también se quede corto con magnitudes similares (+5, +6, +3). No es un error aislado, sino una secuencia con el mismo signo y tamaño parecido. Si, al construir intervalos, se remuestran residuos al azar como si fueran independientes, el muestreo mezcla positivos y negativos y rompe esas rachas (varios pasos seguidos con el mismo signo). Al romperlas, los errores se cancelan con más frecuencia en la simulación que en la realidad y, por tanto, la variabilidad de las trayectorias multi-paso queda subestimada. Esto conduce a intervalos más estrechos de lo que corresponde, cuando en la práctica pueden presentarse varios pasos consecutivos con errores del mismo signo. En escenarios multi-paso, ese remuestreo ingenuo tiende a subestimar la incertidumbre, porque no contempla secuencias realistas de errores encadenados (Figura 66).

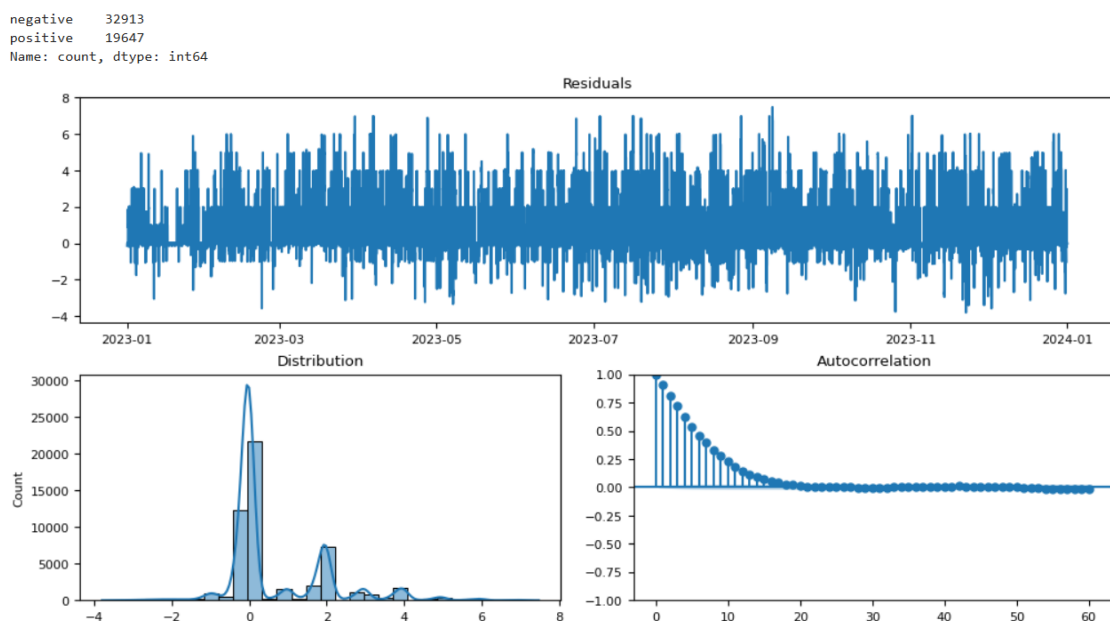


Ilustración 66. Mejor modelo: Residuales off-of-sample

Con este trasfondo, el bootstrapping de residuos OOS (out-of-sample) no condicionado tiene dos caras. Por el lado favorable, al basarse en la distribución empírica de los errores, captura de forma natural la asimetría y las colas pesadas. Por el lado limitante, al ignorar la dependencia serial y el sesgo de signo, puede producir intervalos ligeramente descentrados y optimistas en tramos donde la autocorrelación es fuerte. Aun así, la cobertura global del 81.11% para un objetivo del 80% sugiere calibración razonable en promedio, aunque con posibles desajustes locales por horizonte y periodos de racha.

3.6.3 Métrica y distribución del error por paso del modelo top1: RMdSE_2XGB

Ahora, para reforzar el análisis se examina el error por horizonte de predicción, expresado en pasos de 10 minutos hasta un total de 144 (equivalente a 24 horas) en dos gráficos: métrica del error por paso (Figura 67) y distribución del error por paso (Figura 68).

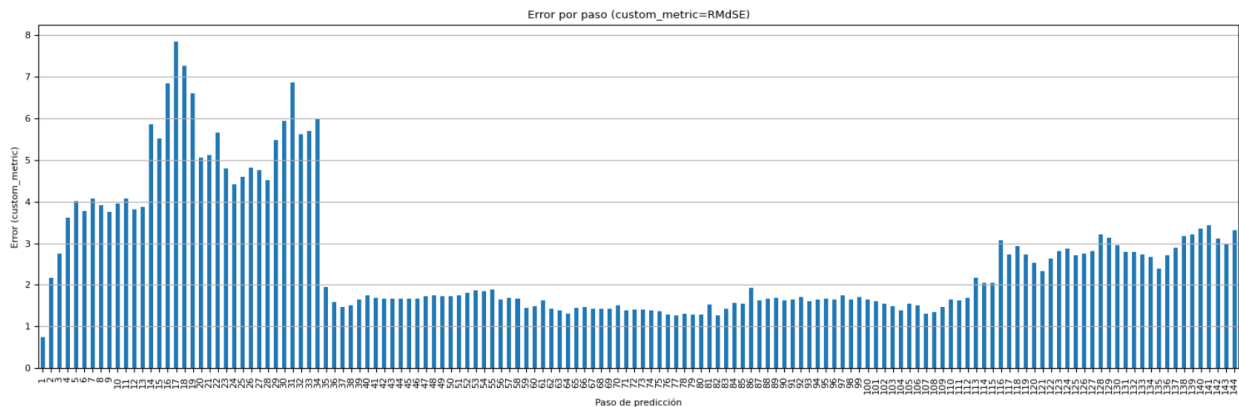


Ilustración 67. Mejor modelo: métrica del Error por paso

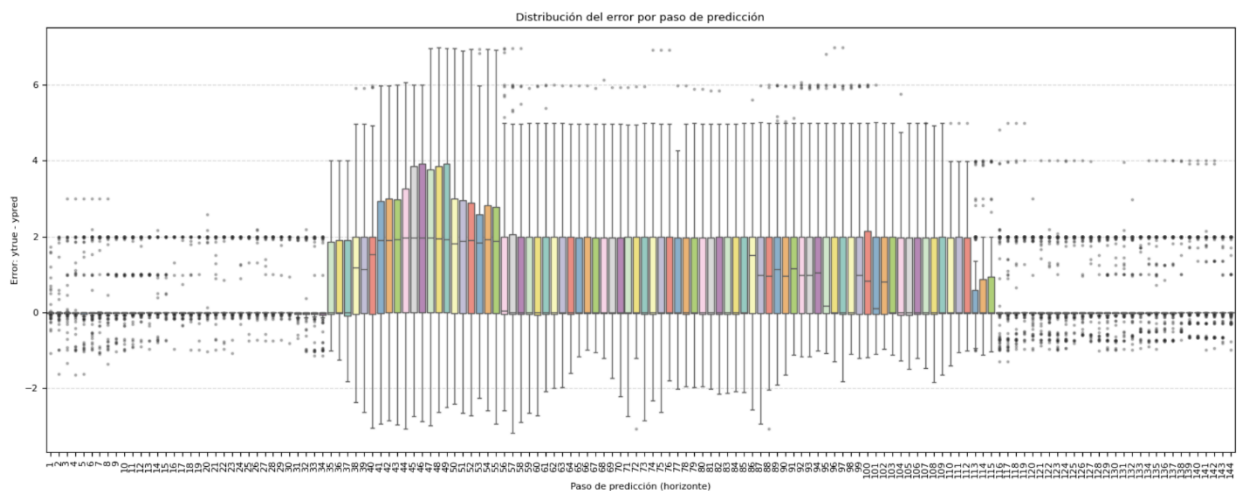


Ilustración 68. Mejor modelo: distribución del Error por paso

El análisis del error por paso mediante una métrica agregada, como el RMdSE, ofrece una visión resumida y cuantitativa del desempeño del modelo en cada horizonte, permitiendo identificar rápidamente dónde se concentran los mayores errores. No obstante, al ser un promedio, oculta detalles sobre sesgo, dispersión o valores atípicos. En contraste, el boxplot complementa este análisis mostrando la distribución completa de los errores, lo que permite detectar tendencias de subestimación, inestabilidad y outliers. En síntesis, la métrica responde: “¿En promedio, qué tan mal me va en cada horizonte?”, mientras que el boxplot responde: “¿Cómo se distribuyen esos errores? ¿Tengo sesgo, dispersión o outliers que debería preocuparme?”.

En la Tabla 14 se muestra en síntesis el comportamiento del modelo, rinde bien en el corto plazo, luego es razonablemente estable, terminano en los últimos pasos con un tramo débil. Sugiriendo considerar ajustes/recalibración para el horizonte de predicción.

Tabla 14. Interpretación de las gráficas errores por paso

Horizonte de predicción	RMdSE por paso (error medio)	Boxplot de errores (distribución)	Interpretación conjunta
Corto plazo (1 a ~10 pasos)	Errores bajos (≤ 2), luego se duplica (≤ 4), tendencia creciente inicial	Cajas compactas, mediana cercana a 0, pocos outliers	Predicciones medianamente confiables y estables en el corto plazo
Temprano–intermedio (~10 a ~35 pasos)	Zona crítica con un error máximo 7–8 unidades	Cajas compactas, mediana cercana a 0, pocos outliers	En horas de la madrugada existe comportamientos atípicos en la demanda
Intermedio estable (~35 a ~115 pasos)	Error bajo y relativamente estable (1–2 unidades)	Medianas desplazadas hacia valores positivos, dispersión muy amplia y presencia de outliers	A partir de las 6am aproximadamente el modelo posee un error ~2 en el rango intercuartílico
Largo plazo (~115 a ~144)	Error en aumento progresivo (hasta ~3.5 unidades)	Cajas compactas, dispersión creciente, outliers más frecuentes hacia positivos	El modelo pierde precisión gradualmente y agarra tendencia a subestimar

3.6.4 Explicabilidad del modelo top1: RMdSE_2XGB

SHAP (*SHapley Additive exPlanations*) es un método muy utilizado para explicar los modelos de *machine learning*. SHAP summary plot ofrece una visión global de cómo cada característica influye en las predicciones del modelo en todo el conjunto de datos. Las variables con mayor impacto global se ubican arriba. En las abscisas, el valor SHAP indica cuánto sube (positivo) o baja (negativo) la predicción en magnitud, de existir rojos y azules en ambos lados, hay efectos no lineales/interacciones.

En la Figura 69 el modelo es fuertemente autoregresivo (“inercia” del dato inmediatamente anterior) a 1 paso (lag_1/inercia/momentum inmediato), en lag_6, lag_12, lag_72 (≈ 1 h, 2 h y 12 h) tienen impacto medio: el modelo usa memoria de corto/medio plazo para ajustar nivel y tendencia intradía. Esto quiere decir que el modelo está impulsado por memoria inmediata (lag_1), nivel reciente (roll_mean_6) y patrón diario (lag_144) que revela la estacionalidad diaria; combina además medidas de volatilidad (roll_std_12) y extremos (roll_max_144) para modular la respuesta. Esta estructura explica por qué el modelo sigue la dirección general, pero subestima los picos en los horizontes críticos.

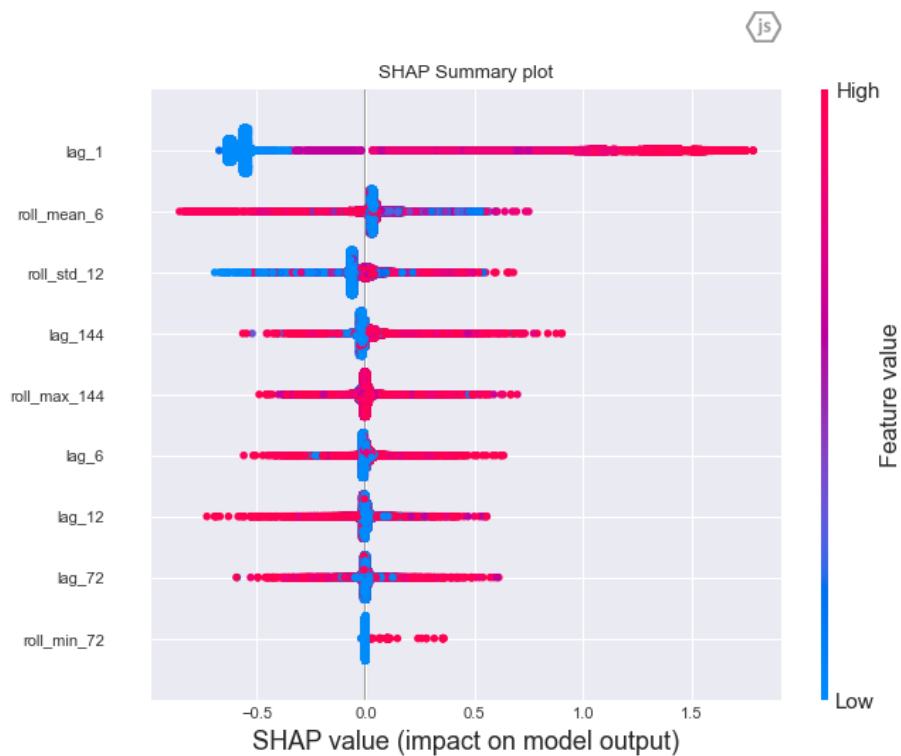


Ilustración 69. Explicabilidad del modelo con SHAP Summary

3.6.5 Trazabilidad y composición del modelo top1: RMdSE_2XGB

En base al modelo top RMdSE_2XGB llamado *UniV_FR C.Pear-P.Impo: Backtesting2 [s_exog_cP_pl]*, grosso modo se explica su composición y trazabilidad (Figura 70).

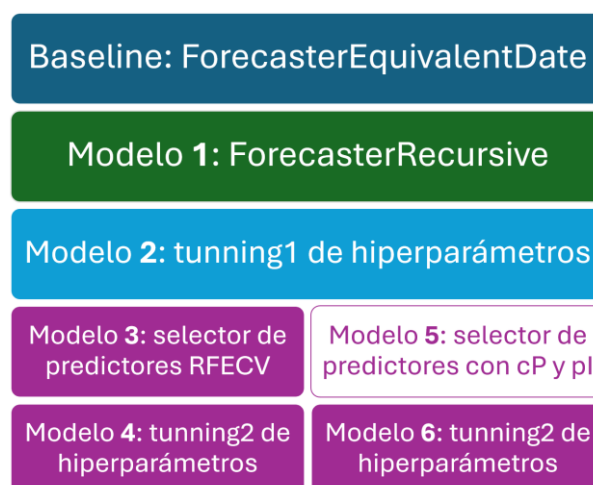
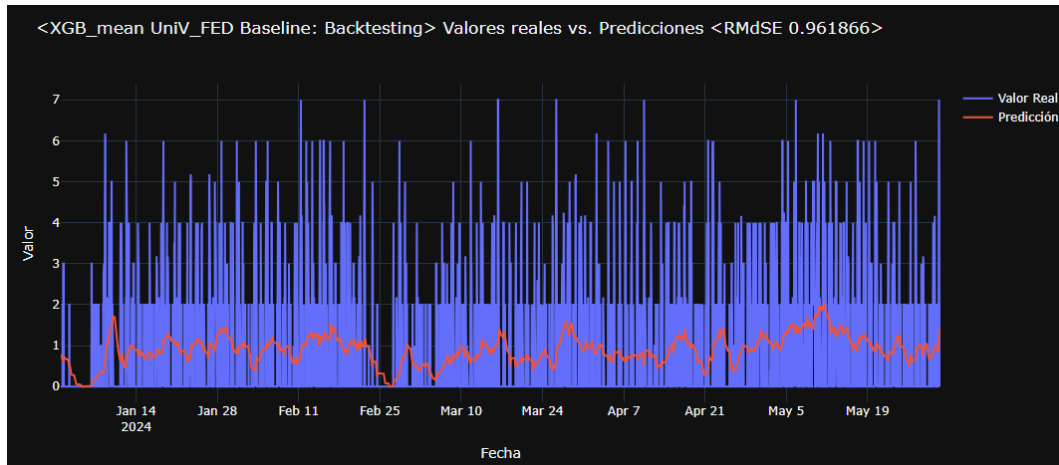


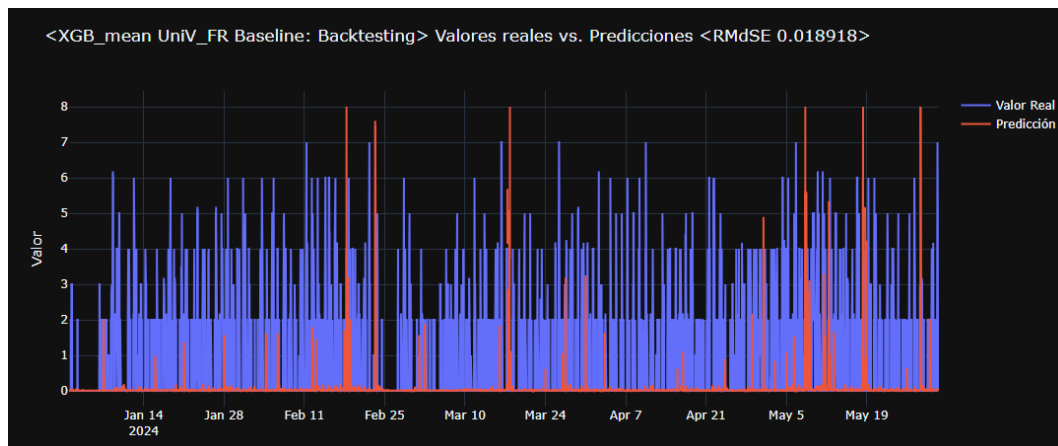
Ilustración 70. Trazabilidad de los modelos explorados para esta investigación

Para una explicación sencilla se creó bloques jerarquía en la Figura 70, estas permiten comprender la trazabilidad del modelo RMdSE_2XGB que está representado por la ilustración por Modelo 5 (bloque contorno lila), el cual se compone de la siguiente forma:

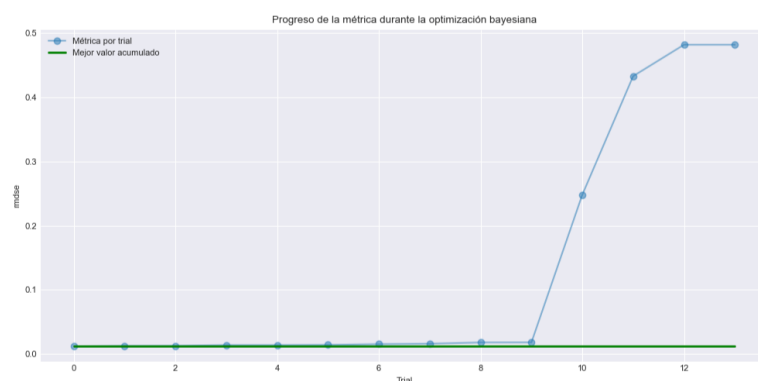
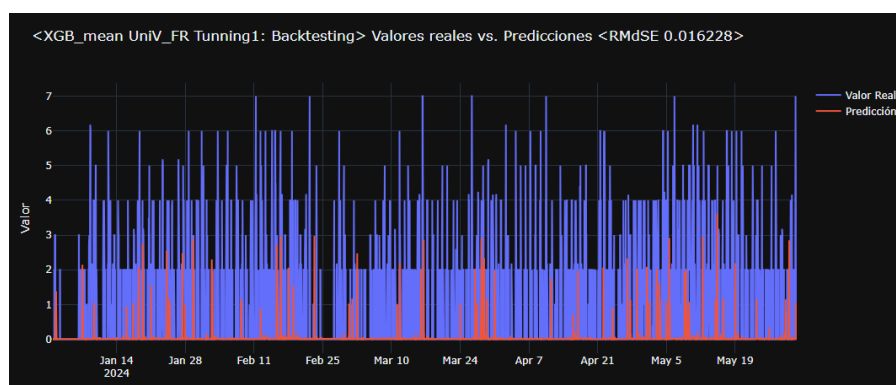
1. Baseline (UniV_FED Baseline: Backtesting): En forecasting, es recomendable disponer de un modelo de referencia simple para iniciar. Se aplica el módulo `ForecasterEquivalentDate` de `skforecast.recursive`, es una estrategia particular de entrenamiento y predicción que busca en el pasado las observaciones equivalentes en fecha al horizonte de predicción, en lugar de usar una regresión normal con lags. Resultados: $RMdSE=0.961866$ y $sMSE=2.445422$.



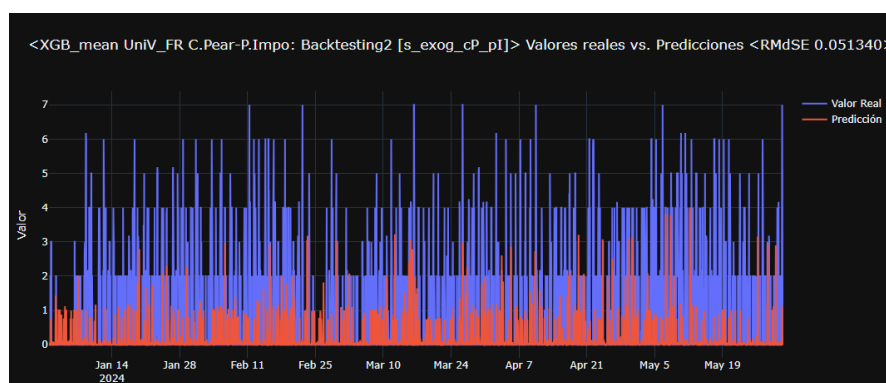
2. Modelo 1 (UniV_FR Baseline: Backtesting): Se construye con el orquestador `ForecasterRecursive` de `skforecast.recursive` con el estimador `XGBRegressor` de Scikit-learn. A partir de aquí se activa la GPU para el regresor. Resultados: $RMdSE=0.018918$ y $sMSE=3.467356$.



3. Modelo 2 (UniV_FR Tunning1: SearchBayesiana + UniV_FR Tunning1: Backtesting): Basado en el anterior Modelo 1 se procede buscar la mejor combinación de lags e hiperparámetros que minimizan un error de predicción como el $RMdSE$, sobre un conjunto de validación. Resultados: $RMdSE=0.016228$ y $sMSE=3.452158$



4. Se pasa por alto Modelo 3 y Modelo 4, ya que estos no intervienen en el modelo top que se está analizando.
5. Modelo 5 (UniV_FR C.Pear-P.Impo: Backtesting2 [s_exog_cP_iP]): En vista que en Modelo 3 sobre RFECV (Recursive Feature Elimination with Cross-Validation) no ayudó al *performance* del modelo (RMdSE=0.067011). Se retoma el orquestador construido por el Modelo2, para buscar otra perspectiva. Luego de un análisis se construye una función que evalúa la relevancia de variables predictoras (exógenas) frente a una variable objetivo combinando dos enfoques: correlación de Pearson para medir relaciones lineales, y la Permutation Importance para cuantificar el impacto real de cada variable en el rendimiento del modelo predictivo. Las variables exógenas de 63 pasan 6 variables con esta función personalizada. Resultado: RMdSE=0.051340.



CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

Este estudio representa un avance en la gestión logística portuaria del Ecuador, al abordar una dimensión poco explorada en la literatura: la planificación de servicios auxiliares en puertos de hidrocarburos. Su enfoque basado en inteligencia artificial constituye una contribución original que aporta nuevas perspectivas a las investigaciones en entornos portuarios estratégicos.

Al aplicar técnicas de *Machine Learning* para anticipar la demanda de remolcadores, se explora una solución innovadora frente a la ausencia de herramientas tecnológicas actuales en el caso de estudio. Este enfoque responde a una problemática crítica, ya que la logística operativa portuaria depende de múltiples factores inciertos, comprometiendo la planificación del servicio y costos asociados. Esta propuesta sienta el inicio de las bases para una toma de decisiones más informada y proactiva, corresponde a la línea de investigación Eficiencia energética, Energías renovables y Alternativas, perteneciente al área de Logística y Transporte del país.

CONCLUSIONES

El propósito del trabajo consistió en realizar un análisis descriptivo y predictivo para el abastecimiento de remolcadores en maniobras de traspaso de hidrocarburos en un terminal portuario marítimo, integrando técnicas de *Machine Learning* bajo un enfoque univariante y multistep. Los resultados se sustentan en una rigurosa exploración estadística y visual de la serie temporal, que permitió comprender sus componentes estructurales —como estacionalidad, intermitencia y leve tendencia— y guiar adecuadamente el modelado.

Respecto al primer objetivo, se identificaron variables endógenas y exógenas: como endógenas, la demanda de remolcadores a 10 minutos modelada con retardos [1, 6, 12, 72, 144] capturando la memoria de la serie, y el resumen del nivel y la variabilidad con window features [roll_mean_6, roll_std_12, roll_min_72, roll_max_144]; y como exógenas, con la función personalizada (Correlación de Pearson y Permutation Importance) evaluó la relevancia de variables predictoras pasando de 63 variables a 6 variables: temporales [hour_cos, hour_sin, hour, week, is_daylight] y meteorológicas [temp_window_12H_max]. La selección se

realizó con criterios técnicos coherentes con el dominio y la estructura temporal de los datos. En relación con el segundo objetivo, el análisis exploratorio reveló un comportamiento altamente intermitente (68% de ceros), estacionalidad diaria significativa, y una débil tendencia positiva, entre otros. Para el tercer objetivo, se escogió modelos de aprendizaje supervisado basados en árboles de decisión —específicamente con *Gradient Boosting Machines*— por su fortaleza en herramientas que emulan el comportamiento en entorno operativos reales, tales como *backtesting* y validación cruzada mediante *TimeSeriesFold*. Finalmente, en cumplimiento del cuarto objetivo, se determinaron métricas adecuadas al contexto (RMdSE y sMSE) basadas en recomendaciones de literatura especializada como Hewamalage *et al.* (2023), excluyendo aquellas inadecuadas para datos intermitentes con ceros. Con ello, se identificó el modelo, proporcionando una validación robusta.

LIMITACIONES

Una limitación fue la aplicación de características polinómicas derivadas de interacciones entre variables exógenas, estas a menudo no actúan de forma independiente; su impacto en la variable objetivo puede depender de otras variables exógenas. Por ejemplo, el efecto de la temperatura sobre la demanda eléctrica varía según la hora del día. Estas interacciones se pueden crear aplicando combinaciones polinómicas de grado 2 con el fin de capturar relaciones no lineales entre variables y mejorar el rendimiento de modelos predictivos. Sin embargo, no se aplicó esta técnica debido a las limitaciones en la capacidad de cómputo, ya que el número de combinaciones en ocasiones superaba más de 300 variables, lo cual al momento de aplicar `fit()`, `backtesting_forecaster()` o `bayesian_search_forecaster()`, sobrepasaba la capacidad de memoria disponible —6 GB— impidiendo continuar con los procesos.

El modelo desarrollado no alcanzó una generalización aceptable a lo largo del horizonte completo. Aunque los intervalos al 80% lograron una cobertura cercana al nominal ($\approx 81.11\%$), las métricas de precisión mostraron un ajuste insuficiente y el análisis de errores evidenció subestimación sistemática, junto con autocorrelación, asimetría, colas pesadas y heterocedasticidad en los residuos. Estas limitaciones son consistentes con las características de la serie (frecuencia

de 10 minutos, alta variabilidad intradía, picos abruptos y posibles cambios de régimen), que el modelo no logró capturar de forma estable, especialmente en horizontes intermedios y largos.

Esto sugiere que, si bien se ha logrado una mejora en eficiencia de cobertura, la precisión absoluta de las predicciones continúa siendo un desafío, y podría beneficiarse de técnicas de modelado fiables como el orquestador ForecasterDirect, a pesar del poder computacional que este demanda.

RECOMENDACIONES

Fortalecer la generalización del modelo con horizontes cortos (7 o 15 días) agregando datos dinámicos (AIS/VHF) para reforzar la ingeniería de características.

Como línea futura, explorar con modelos econométricos y Deep Learning para capturas dependencias complejas no lineales, con monitoreo y actualización ante cambios de régimen. Además, plantear modelos prescriptivos que sugieran acciones concretas frente a escenarios de demanda. También sería valioso escalar esta solución a otros terminales del país, fortaleciendo la eficiencia operativa en el sistema portuario petróleo nacional.

BIBLIOGRAFÍA

- Amat Rodrigo, J., & Escobar Ortiz, J. (2025). skforecast (Version 0.16.0) [Computer software]. <https://doi.org/10.5281/zenodo.8382788>
- Atak, Ü., Kaya, T., & Arslanoğlu, Y. (2022). Analysing the Effects of Weather Conditions on Container Terminal Operations Using Machine Learning [Análisis de los efectos de las condiciones meteorológicas en las operaciones de las terminales de contenedores mediante aprendizaje automático]. *International Journal of Business Analytics (IJBAN)*, 9(5), 1-17. <http://doi.org/10.4018/IJBAN.298016>
- Borzyszkowski, Adrian (2020). *Predicting Port Tugboat Operations for Arriving and Departing Vessels Using Machine Learning. Predicción de operaciones de remolcadores portuarios para la llegada y salida de buques mediante aprendizaje automático* [Tesis de Master]. University of Turku. <https://doi.org/10.1177/0361198120916734>
- Bucheli Tapia, Enrique Ramiro (2014). La disponibilidad y capacidad de operación de las embarcaciones del Terminal Petrolero de La Libertad y su influencia en la recepción y despacho de buques mercantiles en el período 2011-2013 [Tesis de Postgrado]. Universidad Laica Eloy Alfaro. Manta – Ecuador. Disponible en: <https://repositorio.uleam.edu.ec/bitstream/123456789/1368/1/ULEAM-POSG-AP-0012.pdf>
- Chen, S., Wang, F., Wei, X., Tan, Z., & Wang, H. (2020). Analysis of Tugboat Activities using AIS Data for the Tianjin Port [Análisis de las actividades de remolcadores utilizando datos AIS para el puerto de Tianjin]. *Transportation Research Record*, 2674(5), 498-509. <https://doi.org/10.1177/0361198120916734>
- Cryer, Jonathan D.; y Chan, Kung-Sik (2008). Time Series Analysis. With Applications in R (Second Edition). Springer.
- Fayyad, U., Piatetsky-Shapiro, G., y Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17(3), 37. <https://doi.org/10.1609/aimag.v17i3.1230>
- Feifei Yu, Fei Teng, Qihe Shan, Tieshan Li, Yang Xiao (2022). "Continuous Berth Allocation Considering Carbon Emission and Uncertainty" [Un modelo de

- programación robusto para el problema de asignación continua de amarres en condiciones de incertidumbre], 2022 4th International Conference on Data-driven Optimization of Complex Systems (DOCS), pp.1-6.
- Filom, Siyavash; Amiri, Amir M.; Razavi, Saiedeh (2022). Applications of machine learning methods in port operations – A systematic literature review [Aplicaciones de métodos de aprendizaje automático en operaciones portuarias: una revisión sistemática de la literatura]. *Transportation Research Part E: Logistics and Transportation Review*, 161(1). Disponible en: <https://doi.org/10.1016/j.tre.2022.102722>
- Gneiting, T., & Raftery, A. E. (2007). *Strictly proper scoring rules, prediction, and estimation*. *Journal of the American Statistical Association*, 102(477), 359–378. Disponible en: <https://doi.org/10.1146/annurev-statistics-062713-085831>
- Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and practice* (3rd ed.). OTexts. <https://otexts.com/fpp3/>
- Jiahuan Jin, Mingyu Ma, Huan Jin, Tianxiang Cui, Ruibin Bai (2023). Container terminal daily gate in and gate out forecasting using machine [Pronóstico diario de entrada y salida de la terminal de contenedores mediante métodos de aprendizaje automático]. *Transport Policy*, 132(1), pp. 163-174. Disponible en: <https://doi.org/10.1016/j.tranpol.2022.11.010>
- Kim, Kwang Il; y Lee, Keon Myung (2019). Data-Driven Prediction of Ship Destinations in the Harbor Area Using Deep Learning [Predicción basada en datos de los destinos de los barcos en la zona portuaria mediante aprendizaje profundo]. In: Lee, W., Leung, C. (eds) *Big Data Applications and Services 2017. BIGDAS 2017. Advances in Intelligent Systems and Computing, vol 770*. Springer, Singapore. https://doi.org/10.1007/978-981-13-0695-2_10
- Kolley, Lorenz; Rückert, Nicolas; Kastner, Marvin; Jahn, Carlos; y Fischer, Kathrin (2023) Robust berth scheduling using machine learning for vessel arrival time prediction [Programación robusta de atraques mediante aprendizaje automático para predecir la hora de llegada de los buques]. *Flex Serv Manuf J* 35, 29–69. <https://doi.org/10.1007/s10696-022-09462-x>
- Liu, C., Xiang, X. & Zheng, L. (2017). Two decision models for berth allocation problem under uncertainty considering service level [Dos modelos de decisión para el

problema de asignación de ataques en condiciones de incertidumbre considerando el nivel de servicio]. *Flex Serv Manuf J* 29, 312–344. <https://doi.org/10.1007/s10696-017-9295-5>

Obando Larrea, Kevin Andrés (2020). Propuesta de Medidas Preventivas para los Factores de Riesgos Físicos y Mecánicos del Área Operativa de la Superintendencia del Terminal Petrolero de Balao [Tesis de Postgrado]. Pontificia Universidad Católica del Ecuador. Ciudad Esmeralda, Ecuador. Disponible en: <https://repositorio.puce.edu.ec/server/api/core/bitstreams/a52574e9-d24c-4f5e-a0a9-d5e13cbd6f2a/content>

Sepp Hochreiter; y Jürgen, Schmidhuber (1997). Memoria a largo plazo. *Computación neuronal*, 9 (8): 1735–1780. Disponible en: <https://doi.org/10.1162/neco.1997.9.8.1735>

Siddiqui, M. Hameed; y Rodricks, Franklin (2010). "Effective Crude Oil Export Sales Management Towards a Robust Economy" [Gestión eficaz de las ventas de exportación de petróleo crudo hacia una economía sólida]. Paper presented at the SPE/DGS Saudi Arabia Section Technical Symposium and Exhibition, Al-Khobar, Saudi Arabia, April 2010. doi: <https://doi.org/10.2118/136947-MS>

Suillin A (2017) kaggle-web-traffic. Consultado: 19/11/2018. <https://github.com/Arturus/kaggle-web-traffic/>

Wu, Yangcan; y Miao, Lixin (2020). A Robust Scheduling Model for Continuous Berth Allocation Problem under Uncertainty [Un modelo de programación sólido para el problema de asignación continua de ataques en condiciones de incertidumbre]. *2020 5th International Conference on Electromechanical Control Technology and Transportation (ICECTT)*, Nanchang, China, pp. 43-49, doi: 10.1109/ICECTT50890.2020.00017