UNIVERSIDAD MIGUEL HERNÁNDEZ

FACULTAD DE CIENCIAS JURÍDICAS DE ELCHE

GRADO EN ADMINISTRACIÓN Y DIRECCIÓNDE EMPRESAS

TRABAJO FIN DE GRADO

Predicción de ventas mediante modelos ARIMA y Prophet: comparación de métodos y buenas prácticas en series temporales.

Curso Académico 2024-2025

ALUMNA/O: MOHAMED HADDOUTI BAKKALI

TUTOR/A: ÁNGEL SÁNCHEZ BARBIÉ

Índice general

Índice de ilustraciones

1.- RESUMEN / ABSTRACT

Este trabajo analiza y compara diferentes metodologías para la predicción de ventas en series temporales, centrándose en los modelos ARIMA (en sus versiones manual y automática) y Prophet. A partir de un conjunto de datos reales de ventas, se realiza una transformación y exploración inicial para identificar tendencias y patrones de variabilidad. Se llevan a cabo procesos de preprocesamiento, incluyendo la diferenciación y la transformación logarítmica, con el objetivo de adaptar la serie a los supuestos requeridos por los modelos clásicos de predicción.

Se divide la serie temporal en conjuntos de entrenamiento y test, evaluando la capacidad predictiva de los modelos sobre el tramo de datos no visto durante el ajuste. Además, se estudia el impacto de la granularidad temporal sobre la precisión del modelo Prophet, aplicándolo tanto a la serie diaria como a la serie agregada semanalmente. Para cada modelo y configuración, se calculan métricas objetivas de error, como el MAPE, MAE y RMSE, lo que permite realizar una comparación cuantitativa de resultados.

Los hallazgos muestran que la agregación semanal de los datos puede mejorar notablemente la precisión de los modelos de tipo estructural como Prophet, mientras que ARIMA ofrece un rendimiento competitivo en la serie diaria. Finalmente, se discuten las implicaciones prácticas de estos resultados para la toma de decisiones empresariales y se plantean posibles líneas de mejora y extensión futura del trabajo.

1.1 - Objetivo del Proyecto

El objetivo principal de este proyecto es analizar, comparar y evaluar distintas metodologías de predicción aplicadas a una serie temporal de ventas, con el fin de identificar el modelo más preciso y adecuado para anticipar la evolución futura de la demanda. Para ello, se implementan y estudian modelos ARIMA (tanto en su variante de ajuste manual como en la automática) y Prophet, aplicados sobre los mismos datos históricos y bajo diferentes configuraciones de preprocesamiento y granularidad temporal.

El propósito es no solo determinar qué modelo ofrece el menor error de predicción sobre un conjunto de datos no vistos, sino también explorar cómo la transformación de los datos (por ejemplo, la diferenciación, el uso de escalas logarítmicas o la agregación semanal) afecta a la capacidad predictiva de los modelos. Los resultados permitirán establecer recomendaciones prácticas para la aplicación real de modelos de forecasting en contextos empresariales, contribuyendo así a una toma de decisiones informada y basada en datos objetivos.

2. - INTRODUCCIÓN / PRESENTACIÓN / FINALIDAD Y MOTIVOS PRESUPUESTO

La anticipación de la demanda y la predicción de ventas se han convertido en elementos esenciales para la sostenibilidad y el crecimiento de las empresas modernas, especialmente en un contexto caracterizado por la globalización, la digitalización y la aceleración de los cambios en los hábitos de consumo. La capacidad de tomar decisiones rápidas, informadas y basadas en datos permite a las organizaciones optimizar la gestión de inventarios, la planificación de la producción y los recursos, además de incrementar la rentabilidad y mejorar la satisfacción del cliente.

Este proyecto surge como respuesta a la problemática central de la gestión empresarial actual: ¿qué modelo y qué estrategias de preprocesamiento permiten anticipar con mayor precisión la evolución de las ventas? La literatura especializada y la experiencia profesional han demostrado que la aplicación rigurosa de modelos de series temporales puede transformar grandes volúmenes de datos históricos en información estratégica para la toma de decisiones (Makridakis et al., 1998; Hyndman & Athanasopoulos, 2021).

La finalidad de este trabajo es diseñar, implementar y comparar distintas metodologías de predicción de ventas aplicadas sobre una serie temporal real, para identificar no solo el modelo más preciso en términos de error, sino también las mejores prácticas en el tratamiento y preprocesamiento de los datos. Se ha seleccionado para ello dos enfoques ampliamente reconocidos: los modelos ARIMA, conocidos por su eficacia en series estacionarias y su capacidad para capturar dependencias temporales complejas (Box et al., 2015); y Prophet, un modelo estructural que destaca por su flexibilidad ante tendencias, estacionalidades y eventos exógenos, especialmente en series no necesariamente estacionarias (Taylor & Letham, 2018).

Una característica fundamental de este trabajo es la **aplicación práctica de la metodología mediante el lenguaje Python**, empleando paquetes y bibliotecas de referencia en el ámbito del análisis de datos y las series temporales:

- **pandas** y **numpy** para la manipulación y transformación eficiente de los datos;
- **matplotlib** y **seaborn** para la visualización exploratoria y el diagnóstico gráfico;
- **statsmodels** y **pmdarima** para la construcción, ajuste y validación de modelos ARIMA, tanto de forma manual como automática;
- **prophet** para el modelado estructural avanzado, el manejo de estacionalidades múltiples y la incorporación de eventos externos.

El proceso metodológico seguido abarca desde la carga y limpieza de los datos, pasando por el análisis exploratorio, la transformación y preprocesamiento (incluyendo diferenciación, transformación logarítmica y agregación semanal), la división de la serie en conjuntos de entrenamiento y test, el ajuste y validación de modelos bajo distintos enfoques, hasta la evaluación cuantitativa de resultados mediante métricas estándar como MSE, RMSE, MAE y MAPE.

El proyecto no se limita a comparar modelos, sino que estudia el impacto de las distintas estrategias de transformación y granularidad temporal, analizando cómo estas decisiones técnicas pueden mejorar significativamente la precisión y la interpretación de los modelos. Cuando hablamos de **Granularidad temporal** se refiere al nivel de detalle con el que se registran y analizan los datos a lo largo del tiempo en una serie temporal. En otras palabras, indica la unidad de tiempo que se utiliza para medir y representar las observaciones: por ejemplo, diario, semanal, mensual, trimestral, etc.

Una serie con granularidad diaria contiene un dato por cada día; una con granularidad semanal, un dato por semana, y así sucesivamente. La elección de la granularidad depende tanto de cómo se recogen los datos como de los objetivos del análisis y de las decisiones que se desean tomar con las predicciones.

En el contexto de la predicción de ventas y gestión empresarial, la granularidad temporal es un elemento clave por varias razones:

- Determina el equilibrio entre detalle y ruido: granularidades más finas (diarias) proporcionan más información, pero también más variabilidad y ruido; granularidades más agregadas (semanales o mensuales) suavizan la serie y facilitan la detección de tendencias y estacionalidades.
- Afecta a la precisión del modelo: como se ha demostrado en este trabajo, la agregación semanal de los datos permitió a Prophet alcanzar un MAPE significativamente menor que con datos diarios, al eliminar fluctuaciones irrelevantes para la planificación.
- Debe alinearse con las necesidades de decisión: para decisiones operativas de corto plazo (gestión diaria de inventario), puede ser preferible mantener la granularidad diaria; para decisiones tácticas o estratégicas (planificación de producción o campañas), suele ser más útil trabajar con granularidades agregadas como la semanal o mensual.

Esta aproximación es especialmente relevante en la práctica empresarial, donde las series presentan tendencias, ciclos y ruido, y donde la correcta selección del preprocesamiento puede marcar la diferencia entre una predicción útil y otra irrelevante.

Otro concepto que hay que comprender para poder entender el análisis que se lleva a cabo en este proyecto es el de **agregación temporal.**

La **agregación temporal** es el proceso mediante el cual los datos de una serie temporal se combinan o suman a un nivel de detalle mayor al original, cambiando su granularidad. Por ejemplo, transformar una serie de ventas diarias en ventas semanales, mensuales o trimestrales mediante la suma o promedio de los valores correspondientes a cada intervalo de tiempo.

En otras palabras, mientras que la **granularidad** hace referencia a la unidad de tiempo en la que los datos están expresados (diaria, semanal…), la **agregación** es la operación que convierte los datos de una granularidad fina en una más gruesa.

La agregación temporal es muy útil por las siguientes causas:

- **Reduce el ruido:** Datos diarios pueden contener fluctuaciones pequeñas (ruido) irrelevantes para la decisión empresarial. La agregación suaviza la serie, facilitando la identificación de tendencias y estacionalidades.
- **Mejora la precisión:** Al reducir la varianza aleatoria, los modelos pueden ajustarse con mayor precisión a la señal subyacente.
- **Alinea los datos con el objetivo:** Si las decisiones se toman a nivel semanal o mensual, es más coherente trabajar con una serie agregada a esa frecuencia.
- **Simplifica la interpretación:** Una serie con menos puntos y menos variabilidad es más fácil de entender y comunicar.

En este trabajo, por ejemplo, se comprobó que la agregación semanal de los datos permitió a Prophet alcanzar un MAPE mucho menor que sobre los datos diarios, demostrando que la eliminación del ruido de corto plazo puede mejorar significativamente la calidad de las predicciones cuando la serie se modela para objetivos tácticos o estratégicos.

Los métodos más comúnes de agregación temporal son el de **suma**(cuando la variable es acumulativa: ventas,producción...),**promedio**(ústil cuando interesa una media representativa: temperaturas, precios) y **máximo/mínimo**(cuando se anlaiza riesgo o capacidad.

En última instancia, el propósito de este trabajo es aportar una **guía práctica, reproducible y fundamentada tanto en resultados experimentales como en la literatura especializada**, que ayude a profesionales y empresas a seleccionar la metodología y las herramientas más adecuadas a su contexto real, con una visión integrada que combine el rigor cuantitativo con la aplicabilidad estratégica.

3. - ESTADO DE LA CUESTIÓN Y MARCO TEÓRICO

La predicción de series temporales se ha consolidado como un pilar fundamental en la gestión y la toma de decisiones estratégicas de empresas de todos los sectores. El interés creciente por el forecasting se debe tanto al impacto directo que tiene en la optimización de recursos (inventario, personal, compras) como a su capacidad para anticipar tendencias del mercado y responder a la creciente incertidumbre que caracteriza los entornos empresariales actuales.

3.1 - Evolución histórica del forecasting

Desde mediados del siglo XX, el análisis de series temporales ha sido objeto de estudio y aplicación tanto en la academia como en la industria. Los primeros métodos estadísticos, como la media móvil y el suavizamiento exponencial (Goodwin, 2010), permitieron mejorar las previsiones en industrias manufactureras y minoristas. Sin embargo, la progresiva complejidad de los mercados y la disponibilidad masiva de datos han exigido modelos más sofisticados, capaces de captar relaciones dinámicas, estacionalidades múltiples, efectos de calendario y rupturas estructurales (Lapide, 2006).

3.2 - Modelos clásicos: ARIMA y su vigencia.

En este contexto, los modelos ARIMA (AutoRegressive Integrated Moving Average) han sido durante décadas la referencia para la predicción de series temporales univariantes (Box et al., 2015). ARIMA se fundamenta en la combinación de componentes autorregresivos, integración (diferenciación) y medias móviles, permitiendo capturar la dinámica temporal de series estacionarias y, tras la diferenciación adecuada, de series con tendencia o estacionalidad. Su éxito radica en su flexibilidad, interpretabilidad y su capacidad para ajustarse a una amplia variedad de comportamientos. No obstante, el proceso de identificación y ajuste de los parámetros óptimos (p, d, q) suele requerir experiencia y un análisis riguroso de la serie, apoyándose en herramientas como los gráficos ACF y PACF, el test de Dickey-Fuller y criterios de información como el AIC o el BIC (Makridakis et al., 2018).

La automatización de estos procesos ha dado lugar al desarrollo de paquetes y algoritmos que simplifican la selección de modelos, como pmdarima (Tashman & Leach, 1991). Sin embargo, sigue siendo necesaria una validación rigurosa sobre datos no vistos para evitar el sobreajuste y garantizar la robustez de las predicciones (Ben Taieb et al., 2014).

3.3 - Modelos estructurales y Prophet.

La creciente demanda de modelos adaptativos, interpretables y capaces de gestionar grandes volúmenes de datos ha propiciado la aparición de nuevos

enfoques, entre los que destaca Prophet. Desarrollado por Facebook, Prophet se basa en la descomposición aditiva de la serie temporal en componentes de tendencia, estacionalidad y efectos de eventos o festivos (Taylor & Letham, 2018). A diferencia de ARIMA, Prophet no exige que la serie sea estacionaria, permitiendo trabajar directamente con datos con tendencia y estacionalidad múltiple.
El modelo ofrece una configuración flexible, permitiendo añadir estacionalidades personalizadas, ajustar la sensibilidad a cambios de tendencia ("changepoints") y gestionar valores atípicos o ausencias de datos de manera robusta. Por su estructura y facilidad de uso, Prophet se ha popularizado rápidamente tanto en la industria como en la investigación aplicada (Fildes et al., 2019).

3.4 - Tendencias actuales y desafíos emergentes

En los últimos años, el auge del big data y la inteligencia artificial ha ampliado el espectro de técnicas disponibles para el forecasting, incluyendo modelos híbridos, machine learning y redes neuronales recurrentes (Makridakis et al., 2018; Ben Taieb et al., 2014). Sin embargo, los modelos estadísticos clásicos y estructurales como ARIMA y Prophet siguen siendo preferidos en muchas aplicaciones empresariales por su interpretabilidad, bajo requerimiento computacional y buena relación entre complejidad y precisión.

Diversos estudios han puesto de manifiesto la importancia crítica del **preprocesamiento de los datos** para el éxito del forecasting. Prácticas como la transformación logarítmica, la diferenciación para lograr la estacionariedad, la gestión de valores atípicos y, muy especialmente, la **agregación temporal** (por ejemplo, de diario a semanal o mensual) pueden mejorar de forma sustancial la precisión y la robustez de los modelos, ayudando a reducir el efecto del ruido y a potenciar los patrones relevantes (Ghodsi et al., 2021; Prophet Documentation).

3.5 - Evaluación y métricas de desempeño.

La evaluación objetiva de los modelos de predicción exige el uso de métricas robustas y adaptadas al contexto empresarial. Además del error cuadrático medio (MSE) y el error absoluto medio (MAE), se ha extendido el uso del error absoluto porcentual medio (MAPE) por su fácil interpretación y su capacidad para comparar el rendimiento en series de diferentes magnitudes (Hyndman, 2006). El uso de conjuntos de test independientes (holdout o backtesting) es considerado una buena práctica para garantizar la validez de las conclusiones y evitar el sobreajuste.

4. - FUNDAMENTOS y DESCRIPCIÓN TEÓRICA DE ARIMA y PROPHET.

ARIMA

4.1 - Concepto y antecedentes.

ARIMA (AutoRegressive Integrated Moving Average) es una clase de modelos estadísticos introducidos por Box y Jenkins (1970) y perfeccionados en sucesivas décadas. Su popularidad deriva de su capacidad para modelar, a partir de una serie histórica univariante, la relación entre los valores pasados y presentes de la variable de interés, capturando tanto la **inercia** (memoria de la serie) como el impacto de **errores aleatorios** previos, siempre bajo la condición de que la serie sea o se pueda transformar en **estacionaria**.

Una **serie temporal estacionaria** es aquella cuyos **atributos estadísticos fundamentales** como la media, la varianza y la autocorrelación **permanecen constantes a lo largo del tiempo**. En otras palabras, en una serie estacionaria, el comportamiento del proceso no depende del momento en que se observe:

- **La media** es constante.
- **La varianza** (dispersión respecto a la media) es constante.
- **La autocorrelación** entre los valores de la serie depende solo de la distancia entre ellos (lag/rezagos), no del momento específico.

4.2 - Componentes matemáticos

El modelo ARIMA(p,d,q) se compone de tres elementos básicos:

- **AR (AutoRegresivo)**: el valor actual depende linealmente de los p valores previos de la serie.
- **I (Integrated)**: la serie se diferencia d veces para hacerla estacionaria, eliminando tendencias o cambios de nivel.
- **MA (Moving Average)**: el valor actual depende linealmente de los q errores de predicción pasados.

A) Componente autorregresivo (AR(p))

El **componente autorregresivo** (AR) del modelo ARIMA captura la **relación entre el valor actual de la serie y sus propios valores pasados**. Es decir, asume que lo que ocurre hoy depende, en parte, de lo que ocurrió en días, semanas o periodos anteriores.

Matemáticamente, el modelo autorregresivo de orden p, AR(p), se expresa así:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t$$

- $\boldsymbol{y_t}$**:** Valor de la serie temporal en el momento actual t.
- $\boldsymbol{\phi_1, \phi_2, \ldots, \phi_p}$**:** Coeficientes autorregresivos. Miden cuánto influye cada valor pasado.
- $\boldsymbol{y_{t-1}, y_{t-2}, \ldots, y_{t-p}}$ **:** Valores de la serie en los p periodos anteriores.
- $\boldsymbol{\varepsilon_t}$: Error aleatorio (también llamado "ruido blanco"), que representa todo lo impredecible o no explicado por los valores pasados.

Interpretación:
Si, por ejemplo, el coeficiente ϕ1 es grande, eso significa que el valor de ayer tiene mucho peso para predecir el de hoy. Si hay muchos valores de ϕ cercanos a cero, el pasado tiene poca influencia, y el proceso es más random/aleatorio.

B) Componente de media móvil (MA(q))

El **componente de media móvil** (MA) explica el valor actual de la serie como una **combinación lineal de los errores de predicción cometidos en los periodos anteriores**. No depende directamente de los valores previos de la serie, sino de lo "mal" que predijimos en el pasado.

Matemáticamente, un modelo MA(q) de orden q se define así:

$$\boldsymbol{y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}}$$

- $\boldsymbol{\varepsilon_t}$**:** Error aleatorio actual (lo que no explica el modelo).
- $\boldsymbol{\theta_1, \theta_2, \ldots, \theta_q}$**:** Coeficientes de media móvil. Miden la influencia de los errores pasados.
- $\boldsymbol{\varepsilon_{t-1}, \varepsilon_{t-2}, \ldots, \varepsilon_{t-q}}$**:** Errores cometidos en los periodos anteriores.

Interpretación:
Imagina que ayer predijimos mal las ventas. Es probable que ese error afecte tu predicción de hoy. El modelo aprende a corregir estos errores pasados para ajustar la predicción actual.

C) Componente integrado (I(d)): Diferenciación

El **componente integrado** (I) se refiere a la **diferenciación de la serie**, una técnica que consiste en calcular la diferencia entre un valor y el anterior. Su finalidad es eliminar tendencias o patrones de crecimiento/caída a largo plazo y transformar la serie en una **serie estacionaria** (con media y varianza constantes en el tiempo).

- **Primera diferenciación** (d=1):

$$y'_t = y_t - y_{t-1}$$

Esto convierte la serie original en una serie de "cambios" o "incrementos" entre periodos

Segunda diferenciación (d=2), para eliminar tendencias más complejas:

$$y''_t = y'_t - y'_{t-1} = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) = y_t - 2y_{t-1} + y_{t-2}$$

Cada vez que diferenciamos, eliminamos un grado más de tendencia.

- Si hay una tendencia lineal, una diferenciación basta.
- Si hay una tendencia cuadrática (crecimiento acelerado), hacen falta dos.

Lo habitual es que con una o, como máximo, dos diferenciaciones (d=1 o d=2) sea suficiente para lograr estacionariedad en la mayoría de las series temporales económicas o empresariales.

Una diferenciación elimina tendencia lineal; dos, tendencia cuadrática. Más allá de eso, suele indicar que la serie es demasiado ruidosa o que se está intentando eliminar patrones que deberían modelarse de otra manera (por ejemplo, con términos de estacionalidad).

No se recomienda emplear valores de d mayores a 2. En general, la sobre diferenciación tiende a inducir sobreajuste y aumentar la varianza de los errores de predicción.

4.3 - Ajuste automático de modelos ARIMA: el paquete pmdarima

En la práctica, identificar manualmente los mejores parámetros para un modelo ARIMA puede ser un proceso complejo y laborioso, especialmente cuando se dispone de grandes volúmenes de datos o cuando la serie presenta patrones difíciles de interpretar visualmente. Para facilitar y agilizar este proceso, se emplea la biblioteca de Python **pmdarima**, que implementa la función auto_arima, ampliamente reconocida en el entorno de análisis de series temporales.

Pmdarima permite **automatizar la búsqueda y ajuste del modelo ARIMA óptimo** para una serie temporal, explorando de manera sistemática diferentes combinaciones de los parámetros autorregresivos (p), de diferenciación (d) y de media móvil (q), así como componentes estacionales en caso necesario. La selección del modelo final se basa en la minimización de criterios de información estadística como el **AIC** (Akaike Information Criterion), buscando así el equilibrio entre precisión predictiva y simplicidad.

Entre las principales ventajas de pmdarima destacan:

- La reducción del esfuerzo y el conocimiento experto requeridos para seleccionar el modelo adecuado.
- La capacidad para validar de forma objetiva el rendimiento de múltiples configuraciones.
- La integración sencilla con otras herramientas habituales de análisis en Python.

En la **parte experimental** de este trabajo se utilizará pmdarima tanto para **automatizar la selección del mejor modelo ARIMA** como para comparar su rendimiento con el ajuste manual y con otros métodos avanzados de predicción. Esto permitirá evaluar si la automatización puede mejorar la calidad de las predicciones, agilizar el proceso y facilitar su replicabilidad en contextos empresariales reales.

PROPHET

4.4 - Contexto y motivación

Prophet es un modelo de series temporales de tipo **estructural aditivo** desarrollado por Facebook (Taylor & Letham, 2018), especialmente orientado al análisis y la predicción de datos de negocio. Surge como respuesta a las limitaciones de los modelos clásicos (como ARIMA y Holt-Winters) en entornos con **datos incompletos, valores atípicos, estacionalidades múltiples y cambios de tendencia abruptos**. En la práctica empresarial, muchas series temporales presentan estos desafíos, junto con efectos exógenos como días festivos, promociones o eventos excepcionales, que pueden influir significativamente en el comportamiento de la serie.

El principal objetivo de Prophet es **ofrecer un modelo potente pero accesible**, que pueda ser utilizado por analistas y equipos de negocio sin profundos conocimientos estadísticos, permitiendo al mismo tiempo una interpretación sencilla y la personalización de las componentes relevantes para cada caso de uso. Prophet está pensado para situaciones típicas de forecasting de ventas, demanda, tráfico web, reservas, etc., donde los datos muestran tanto estacionalidades (por ejemplo, semanal y anual) como tendencias no lineales, y donde es importante poder ajustar el modelo de manera flexible y entender el impacto de eventos puntuales.

4.5 - Formulación matemática básica y componentes

Prophet modela la serie temporal como la **suma de varios componentes aditivos:**

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t$$

- y(t): valor observado en el instante **t.**
- g(t): componente de **tendencia** (crecimiento lineal o logístico, con posibles puntos de cambio).
- s(t): componente de **estacionalidad** (una o varias, como semanal, anual o personalizada, modelada con series de Fourier).
- h(t): **efectos de festivos o eventos especiales**, opcionalmente añadidos por el usuario.

- ε_t: término de error (ruido blanco).

4.5.1 - Tendencia (g(t))

Prophet permite dos opciones principales para la tendencia:

a) Tendencia lineal segmentada (con puntos de cambio o "changepoints")

$$g(t) = (k + a(t)^{\top}\delta)t + (m + a(t)^{\top}\gamma)$$

- k: pendiente inicial de la tendencia.
- m: intercepto (nivel inicial).
- a(t): vector binario de longitud igual al número de puntos de cambio, que indica en qué tramo de tendencia estamos (1 si t es posterior al changepoint, 0 en caso contrario). Se utiliza la transpuesta para poder multiplicarlo con δ o con γ.
- δ, γ: vectores de saltos de pendiente y nivel para cada punto de cambio ("changepoint").

Esto hace posible que Prophet configure automáticamente cambios de tendencia en la serie (por ejemplo, un cambio de ritmo en el crecimiento de las ventas).

b) Tendencia logística (con saturación):

Cuando el crecimiento de la serie no puede superar un cierto límite (saturación, como en casos de penetración de mercado), Prophet ofrece la opción de tendencia logística:

$$g(t) = \frac{C}{1+\exp\left(-\left((k+a(t)^{\top}\delta)t+(m+a(t)^{\top}\gamma)\right)\right)}$$

donde C es la capacidad máxima.

4.5.2 -Estacionalidad (s(t))

La estacionalidad en Prophet se representa mediante la suma de componentes periódicos que captan los patrones recurrentes (por ejemplo, ciclos semanales, anuales…personalizados).
Esto se modela con **series de Fourier**, permitiendo que la estacionalidad tenga formas complejas, no necesariamente sinusoidales simples.

Fórmula de la estacionalidad:

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

- P: periodo de la estacionalidad (por ejemplo, 7 para semanal, 365.25 para anual).
- N: número de términos de Fourier (determina la complejidad y flexibilidad para captar patrones).
- a_n,b_n: coeficientes ajustados para cada término. Determinan el peso de cada componente seno y coseno.
- $\cos\left(\frac{2\pi nt}{P}\right), \sin\left(\frac{2\pi nt}{P}\right)$: Son las **funciones básicas periódicas** que forman la base de la estacionalidad en Prophet. Se trata de ondas senoidales (una coseno y otra seno) que oscilan con el tiempo t.
- n determina cuantos ciclos temporales caben en P, es decir, si P=365.25(1año), y n= a 1, significa que una onda completa un ciclo en 1 año.
- t representa tiempo, la fechas de nuestros datos expresadas en datos numéricos.

Prophet permite combinar varias estacionalidades simultáneamente (ejemplo: semanal y anual) y personalizarlas según las necesidades del usuario.

4.5.3 - Efectos de festivos y eventos (h(t))

Prophet incorpora una función específica para modelar el impacto de **días festivos, promociones o eventos exógenos** en la serie temporal:

$$h(t) = \sum_{j=1}^{J} \kappa_j D_j(t)$$

- $\boldsymbol{J}$: número total de eventos considerados.
- $\boldsymbol{\kappa_j}$**:** impacto estimado del evento j (puede ser positivo o negativo).
- $\boldsymbol{D_j(t)}$**:** variable indicadora (1 si t corresponde a la fecha del evento j, 0 en caso contrario).

Esto permite incorporar conocimiento experto del negocio, por ejemplo, el efecto de la Navidad, campañas de rebajas, días festivos nacionales o locales, etc.

4.5.4. Término de error (ε_t)

En la formulación de Prophet, el **término de error ($\boldsymbol{\varepsilon_t}$)** representa la parte de la serie temporal que no puede ser explicada por los componentes estructurales del modelo: tendencia, estacionalidad y eventos. Matemáticamente, este término se añade de forma aditiva a la suma de los otros componentes:

$$\boldsymbol{y(t) = g(t) + s(t) + h(t) + \varepsilon_t}$$

Por defecto, Prophet asume que estos errores (ε_t) se comportan como **ruido blanco**, es decir:

- Siguen una **distribución normal** (o gaussiana), es decir, $\varepsilon_t \sim N(0,\sigma 2)$
- **Media cero**: La de los errores a lo largo del tiempo es cero, lo que significa que el modelo no comete un sesgo sistemático ni sobreestima ni subestima la serie de forma consistente.
- **Varianza constante**: Los errores tienen una dispersión constante en el tiempo, una propiedad conocida como homocedasticidad.
- **Independencia temporal**: Los errores no están correlacionados entre sí, es decir, el valor de un error en el tiempo **t** no depende del error cometido en otros periodos.

En la práctica:

Después de ajustar Prophet, es recomendable analizar visualmente los residuos (histograma, gráfico de residuos vs tiempo) y mediante tests estadísticos, para verificar que no existen patrones no modelados. Si se detectan, puede ser necesario añadir más componentes al modelo (por ejemplo, nuevas estacionalidades, eventos, o ajustar el parámetro de sensibilidad a outliers).

4.6 - Ajuste, personalización y uso práctico de Prophet

Prophet realiza la estimación de todos los parámetros de forma automatizada utilizando **máxima a posteriori** (MAP) e inferencia bayesiana, lo que le permite obtener resultados robustos incluso ante series incompletas o con valores

atípicos.
El modelo ajusta automáticamente la posición y número de **changepoints** (puntos de cambio de tendencia), aunque el usuario puede controlar la sensibilidad del modelo a estos cambios mediante el parámetro changepoint_prior_scale.

Además, podemos:

- Activar, desactivar o personalizar las estacionalidades.
- Definir la capacidad máxima para tendencias logísticas.
- Incluir o eliminar efectos de festivos y eventos personalizados.
- Controlar la complejidad de las estacionalidades mediante el número de términos de Fourier.
- Consultar intervalos de confianza para las predicciones, lo que aporta información sobre la incertidumbre inherente a las previsiones.

Prophet también facilita la visualización de los **componentes individuales** (tendencia, estacionalidad, eventos), lo que mejora la interpretabilidad y la comunicación de los resultados en entornos de negocio.

5. - PARTE EXPERIMENTAL: METODOLOGÍA, DESARROLLO y RESULTADOS

5.1 - Introducción y objetivos experimentales.

El objetivo de la parte experimental es analizar, construir y validar modelos de predicción sobre una serie temporal de ventas, utilizando metodologías estadísticas y de machine learning ampliamente reconocidas: **ARIMA** y **Prophet**. El análisis se centra tanto en el proceso de modelado clásico como en la comparación de enfoques manuales y automáticos, y explora cómo la transformación y la granularidad de los datos afectan la capacidad predictiva

5.2 - Análisis exploratorio y transformación inicial.

5.2.1 - Visualización inicial

Se parte de una **serie temporal de ventas diarias** de 400 días, cargada desde un archivo CSV.

La primera acción es graficar la evolución de las ventas, eliminando los ejes verticales para centrar la atención en el patrón visual.

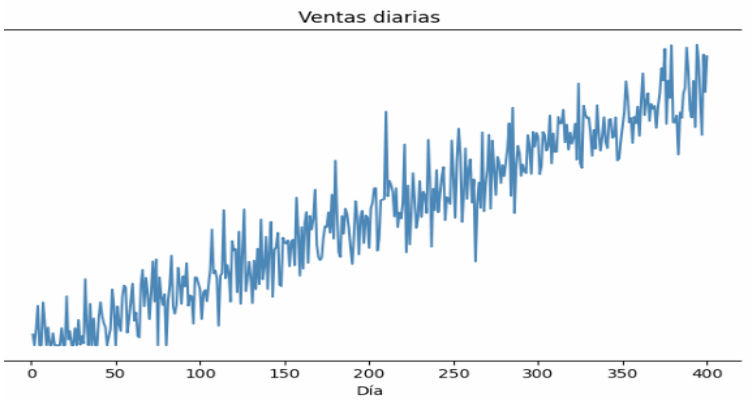

Ilustración 5.0.1: Exploración inicial de la serie temporal

Gráfica de ventas diarias:

En la ilustración 5.0.1 podemos apreciar las tendencias, picos y patrones iniciales, detectando una **tendencia creciente y fluctuaciones notables y constantes en el tiempo**.

5.2.2 - Transformación logarítmica

Se aplica la transformación logarítmica a la serie de ventas por los siguientes motivos:

- **Estabiliza la varianza** (reduce la heterocedasticidad).
- **Linealizar** el crecimiento exponencial.
- Finalmente, permite **interpretar los cambios** en términos relativos (%).

Esta transformación es fundamental en series donde la magnitud de los valores crece con el tiempo, y ayuda a que los modelos estadísticos funcionen de manera óptima. La representación gráfica en escala logarítmica permite detectar con mayor claridad los periodos de crecimiento, los cambios de tendencia y la presencia de patrones recurrentes, preparando así la serie para un modelado más robusto y fiable.

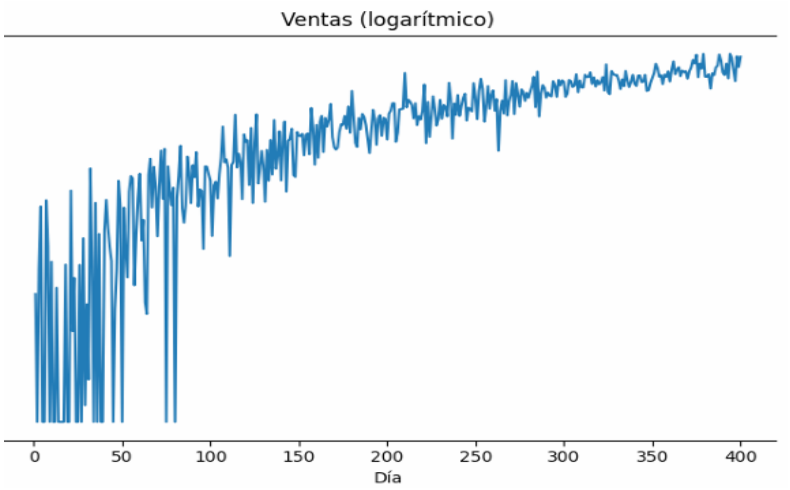

Ilustración 5.0.2: Serie temporal post transformación logarítmica

En la ilustración 5.0.2 podemos observar que el gráfico tiene una menor variación a lo largo del tiempo en comparación al gráfico original.

5.3 - División de la serie: entrenamiento y test

La serie temporal se divide en dos subconjuntos:

- **Entrenamiento:** Todos los días salvo los últimos 30 (para ajustar el modelo).
- **Test:** Los últimos 30 días (para validar la capacidad predictiva).

Este procedimiento ("holdout validation") **simula un escenario real**, evaluando el modelo en datos no vistos.

5.4 - Análisis de estacionariedad y diferenciación.

Ahora se procede a la verificación de la estacionariedad de la serie. El modelo Arima se aplica como bien sabemos en series temporales estacionarias. Esto implica que debemos verificar si las propiedades estadísticas de nuestra serie temporal se mantienen constantes y aproximadamente iguales a lo largo del tiempo. Hay diferentes formas de verificar si una serie temporal es estacionaria:

1. Podemos afirmar que una serie temporal es estacionaria o no, observando su gráfico. Si hay una clara tendencia que se puede observar a simple vista es que no es estacionaria, pero muchas veces la serie es mucho más compleja y no podemos sacar conclusión alguna del gráfico principal, por ello están los 2 siguientes métodos.

2. También podemos determinar la estacionariedad de la serie acudiendo a los gráficos de autocorrelación ACF y autocorrelación parcial PACF. Estos dos gráficos no solo nos van a servir para esta tarea, si no que van a ser útiles durante todo el proceso de ajuste del modelo.

3. Finalmente, podemos determinar si una serie temporal es estacionaria o no mediante el P-valor, o como es conocido también test Dickey Fuller. El test de Dickey-Fuller aumentado evalúa si una serie temporal es estacionaria. Es decir, si sus propiedades estadísticas (media, varianza, covarianza) no cambian a lo largo del tiempo. Este p-valor te está diciendo cuál es la probabilidad de obtener tus datos asumiendo que la serie es no estacionaria (hipótesis nula).

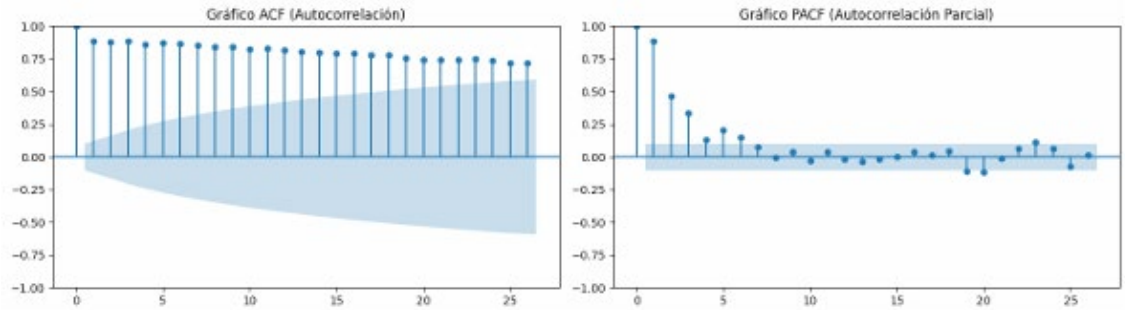

Ilustración 5.0.3: Gráficos ACF y PACF de la serie temporal pre-diferenciación.

5.4.1 - Diagnóstico de estacionariedad.

Gráficos ACF y PACF:

- En el gráfico ACF de la ilustración 5.3 podemos observar que hay barras(lags) altas y consistentes en todos los lags hasta el 25. Estas barras están muy por encima del área azul, que representa los límites de confianza estadística (normalmente al 95%), hay una persistencia fuerte en las correlaciones pasadas. Este patrón es típico de una serie con tendencia, lo que también implica que nos encontramos ante una serie no estacionaria.

- Lo que apreciamos en el **PACF** es que el lag 1 tiene una barra muy alta (muy significativa). A partir del lag 2 en adelante, las barras caen rápidamente y se mantienen dentro de los límites de confianza. Todo lo anterior implica que solo el primer valor pasado (lag 1) aporta información significativa para predecir el presente. Esto es característico de un modelo AR(1): el valor actual depende fuertemente del valor inmediatamente anterior, pero no de los anteriores a ese.

Test de Dickey-Fuller:

- P-valor = 0.938, **mucho mayor que 0.05**. Esto implica que la serie **no es estacionaria**. No se puede rechazar la hipótesis nula, la cual sostiene que la serie temporal en cuestión es no estacionaria.

Por lo tanto, ahora para poder modelar con Arima necesitamos diferenciar los datos para eliminar la tendencia y hacer la serie estacionaria.

5.4.2 - Diferenciación de la serie.

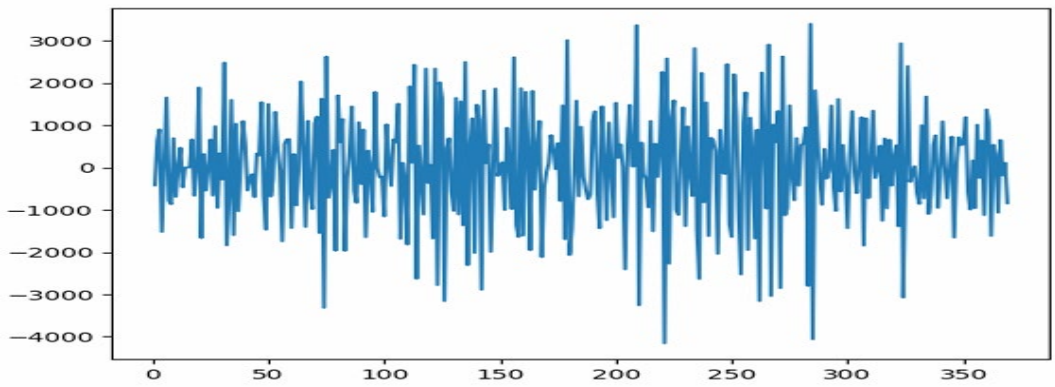

Ilustración 5.0.4: Disposición de la serie temporal post diferenciación

Se realiza la diferenciación de primer orden para eliminar la tendencia:

- Tras diferenciar, la ilustración 5.0.4 nos muestra como la serie fluctúa alrededor de cero. En la gráfica obtenida se observa la serie diferenciada, donde ya no aparece la tendencia creciente de la serie original. En su lugar, los valores fluctúan alrededor de cero, indicando que la serie diferenciada es más estable y posiblemente estacionaria. Este comportamiento es el deseado antes de ajustar un modelo ARIMA, ya que mejora su capacidad para capturar patrones y hacer predicciones fiables.

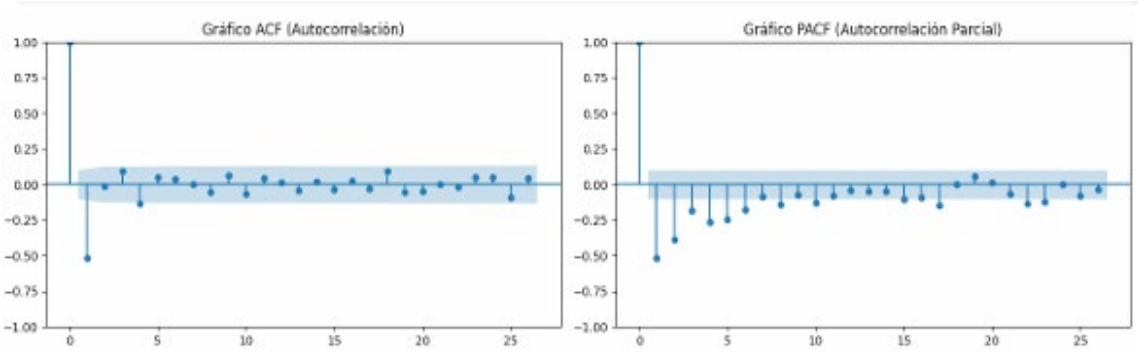

Ilustración 5.0.5: Gráficos ACF y PACF post diferenciación

- En la ilustración 5.0.5 los gráficos ACF y PACF muestran cortes rápidos, confirmando la estacionariedad.

  La autocorrelación ha desaparecido a partir del lag 2, tanto en ACF como PACF. Ya no hay memoria de largo plazo en la serie. La estructura de correlación es simple, corta y débil, justo lo que ocurre en una serie

estacionaria. Visualmente, los puntos oscilan alrededor de cero y dentro del intervalo de confianza

```
adf_test = adfuller(df_train_diff)

print(f"P-Valor: {adf_test[1]: .20f}")
```

```
P-Valor:  0.00000000000046424729
```

Ilustración 5.0.6: Valor del P-valor post difrenciación

- Se verifica la estacionariedad visualmente y con el test ADF (**p-valor < 0.05** tras diferenciar).

5.5 - Identificación y ajuste de modelos ARIMA.

Tras realizar la diferenciación de primer orden para hacer la serie estacionaria (d=1), el siguiente paso es determinar los parámetros p y q del modelo ARIMA. Para ello, se emplean los gráficos de autocorrelación (ACF) y autocorrelación parcial (PACF) (obtenidos tras la diferenciación), siguiendo las reglas generales clásicas:

- Si el PACF muestra un corte claro tras un cierto lag p, mientras que el ACF decae gradualmente, es indicativo de un modelo ARIMA(p, d, 0).

- Si el ACF muestra un corte tras un lag q, mientras que el PACF decae gradualmente, es indicativo de un modelo ARIMA(0, d, q).

En nuestro caso, se observa que el gráfico ACF presenta un pico significativo en el lag 1 y el PACF no corta claramente, lo que sugiere la conveniencia de un modelo ARIMA(0,1,1). Además, la experiencia y la literatura recomiendan que, salvo indicios claros de modelos mixtos, se dé prioridad a modelos simples de tipo AR (autorregresivo puro) o MA (media móvil puro).

No obstante, la identificación de modelos ARIMA basada en ACF/PACF es orientativa y, debido a la posible subjetividad de la interpretación, es recomendable contrastar varias configuraciones para elegir el modelo más adecuado. Por ello, se ha decidido comparar varios modelos ARIMA, concretamente ARIMA(1,1,0), ARIMA(2,1,0), ARIMA(0,1,1), ARIMA(1,1,1) y ARIMA(5,1,0), evaluando su desempeño tanto en el ajuste (mediante el criterio AIC) como en la capacidad predictiva sobre el tramo de test (mediante el error cuadrático medio, MSE).

De este modo, el proceso combina el criterio teórico, basado en el análisis gráfico, con un enfoque empírico que permite seleccionar de forma objetiva el modelo óptimo en función de su rendimiento real.

Los resultados de este análisis comparativo se muestran a continuación:

	modelo	AIC	MSE
0	ARIMA(1, 1, 0)	6241.312997	2.455850e+06
1	ARIMA(2, 1, 0)	6184.898207	2.188916e+06
2	ARIMA(0, 1, 1)	6119.183159	1.871935e+06
3	ARIMA(1, 1, 1)	6118.029144	1.835183e+06
4	ARIMA(5, 1, 0)	6133.223977	1.905514e+06

Ilustración 5.0.7: Criterios AIC y MSE de los modelos

- **El AIC y el MSE**

En la selección y comparación de modelos de series temporales, es fundamental contar con criterios objetivos que permitan identificar la opción más adecuada tanto desde el punto de vista del ajuste como de la capacidad predictiva. Dos de los indicadores más utilizados para este fin son el AIC (Criterio de Información de Akaike) y el MSE (Error Cuadrático Medio).

El **AIC** mide la calidad del ajuste del modelo penalizando la complejidad; es decir, favorece modelos que expliquen bien los datos, pero que sean lo más simples posible. Su fórmula es:

El AIC se calcula como AIC = 2k – 2 ln(L).

La **k** es el número de parámetros del modelo y L es la máxima verosimilitud. El modelo con menor AIC suele ser preferido porque equilibra ajuste y parsimonia, evitando sobreajustes.

MSE, por su parte, mide el promedio de los errores al cuadrado entre las predicciones y los valores reales, pero aplicado sobre los datos de test. Su fórmula es:

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

- $\boldsymbol{n}$**:** Número total de observaciones o predicciones.
- $\boldsymbol{y_i}$**:** Valor real (observado) de la variable en el instante iii.
- $\boldsymbol{\hat{y}_i}$**:** Valor predicho (estimado por el modelo) en el mismo instante iii.

- $(y_i - \hat{y}_i)^2$**:** El **cuadrado de la diferencia** (error) entre el valor real y el valor predicho para cada observación.
- $\sum_{i=1}^{n}$ **:** Suma esos errores al cuadrado para todas las observaciones del conjunto de test.
- $\frac{1}{n}$**:** Calcula el **promedio** de esos errores al cuadrado

Un **MSE** bajo indica que el modelo predice con precisión los valores futuros, y por eso es la métrica estádar para evaluar la capacidad predictiva.

- **Interpretación práctica de los resultados que podemos visualizar en la ilustración 5.0.7**

El modelo ARIMA(1,1,1) es el que presenta el AIC más bajo (6118.03), indicando que es el que mejor equilibra el ajuste a los datos y la simplicidad.

También obtiene el MSE más bajo (1,831,583), lo que significa que es el que mejor predice los valores futuros (los datos de test).

El modelo ARIMA(0,1,1), que la lógica sugería como razonable según el análisis de ACF y PACF, también muestra muy buenos resultados, con un AIC solo ligeramente superior y un MSE muy próximo al del modelo óptimo.

Los modelos con parámetros más altos (como ARIMA(5,1,0)) no mejoran la predicción; de hecho, en este caso, su AIC y MSE son algo peores, lo que ilustra cómo la mayor complejidad no siempre implica mejores resultados y puede conducir a un leve sobreajuste. El uso conjunto de AIC y MSE nos han permitido seleccionar el modelo más adecuado para nuestra serie temporal. La evidencia empírica respalda el uso del modelo ARIMA(1,1,1), que combina un buen ajuste con capacidad predictiva.

No obstante, el modelo ARIMA(0,1,1) también se revela como una opción muy sólida, apoyando el razonamiento inicial extraído de la inspección visual de los gráficos ACF y PACF. Esto pone de manifiesto la importancia de combinar el análisis teórico con la evaluación práctica y cuantitativa en la selección de modelos de series temporales.

Cabe resaltar que AIC y MSE no solo son criterios estandarizados y aceptados en la literatura, sino que, como se ha visto en nuestro análisis, proporcionan una base sólida y objetiva para elegir el modelo de predicción más robusto y fiable.

5.6 - Análisis y validación de los residuos del modelo ARIMA(1,1,1)

Antes de proceder a realizar predicciones con el modelo ARIMA seleccionado, es fundamental comprobar si el modelo ha captado adecuadamente la estructura de la serie temporal. Para ello, se analiza el comportamiento de los residuos, que deberían comportarse como ruido blanco si el modelo es correcto, es decir, deberían ser aleatorios y no contener información sistemática no explicada por el modelo.

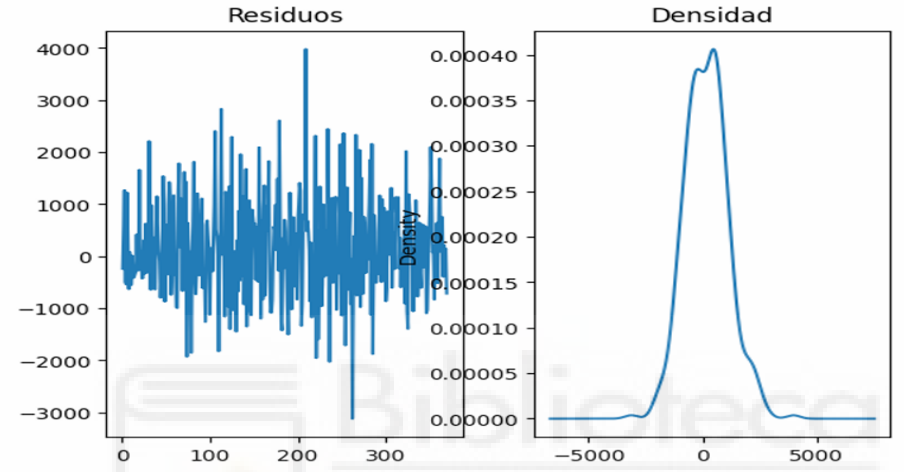

Ilustración 5.0.8: Gráficos de los residuos y de la densidad del modelo

Lo que se observa en la ilustración 5.0.8, es que los **residuos** no presentan patrones identificables ni tendencia aparente. Sus valores oscilan de manera aleatoria alrededor de cero, lo que indica que el modelo ha capturado la información estructural relevante de la serie original.

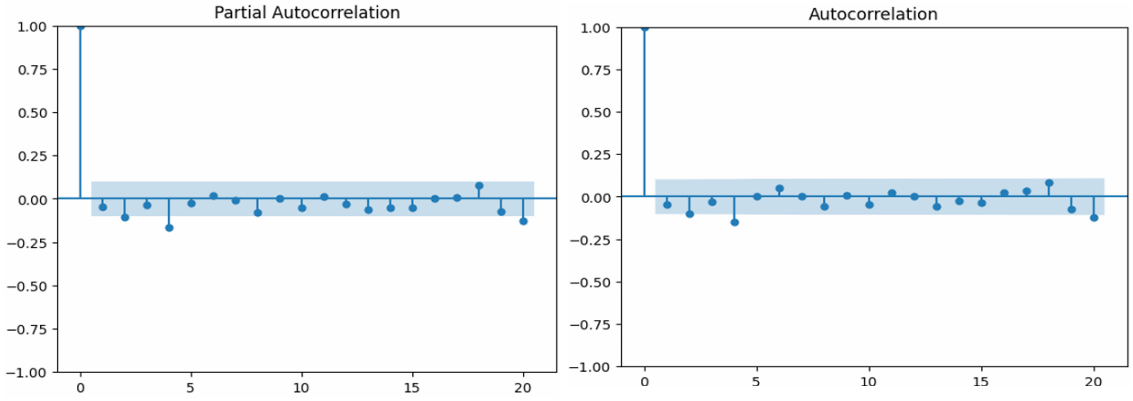

Ilustración 5.0.9: Gráficos ACF y PACF de los residuos

Por otro lado, la **densidad** de los residuos muestra una distribución aproximadamente normal, centrada en torno a cero, lo cual es un buen indicio de que los errores del modelo son aleatorios y no sistemáticamente sesgados.

En la ilustración 5.0.9 se representan los gráficos de autocorrelación (ACF) y autocorrelación parcial (PACF) aplicados a los residuos del modelo.

Se observa que la gran mayoría de los lags se sitúan dentro del intervalo de confianza (zona azul), sin picos significativos. Esto significa que no hay autocorrelación significativa en los residuos: los errores del modelo no dependen de los valores pasados, son generalmente aleatorios, lo que caracteriza un buen modelo de serie temporal.

5.7 - Predicción y visualización ARIMA.

5.7.1 - Predicción manual (ajuste propio).

Se empieza calculando **las predicciones en Python y luego las** graficamos **junto a los datos reales de la serie temporal que hemos analizado.**

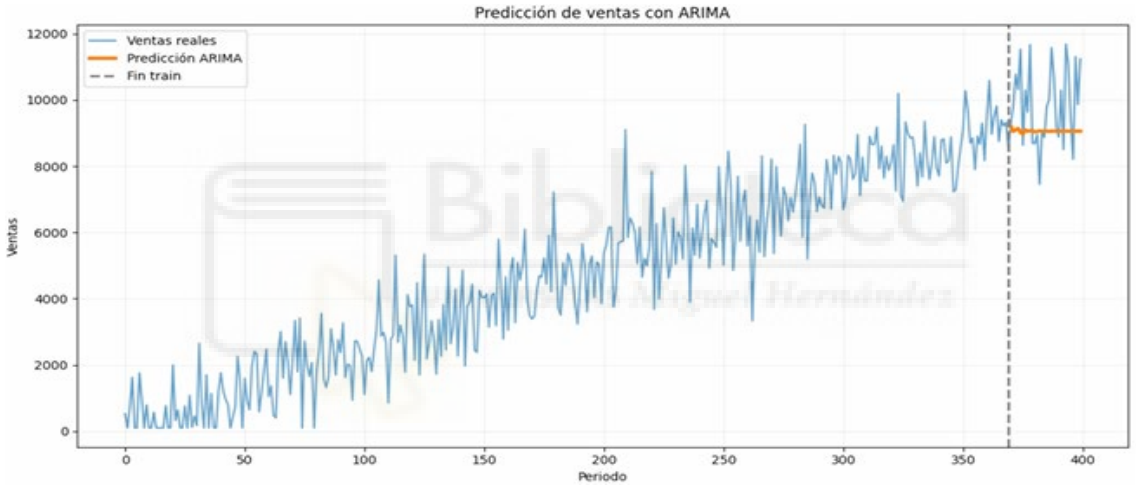

Ilustración 5.0.10: Gráfico de la comparación de las ventas reales con la predicción manual

En el gráfico 5.0.10 se representa la serie temporal de ventas reales junto con las predicciones obtenidas mediante el modelo ARIMA(1,1,1), el cual fue seleccionado tras un proceso exhaustivo de comparación y validación.

La descripción del gráfico Serie azul (“Ventas reales”) representa el histórico completo de la serie temporal (400 observaciones en este caso), mostrando tanto el tramo de entrenamiento como el de test.

Línea vertical discontinua: marca el punto de separación entre el conjunto de entrenamiento y el conjunto de validación (test).

La Línea naranja (“Predicción ARIMA”) indica los valores pronosticados por el modelo únicamente en el tramo de test, es decir, sobre los datos no vistos durante el entrenamiento.

5.7.1.1 - Interpretación y valoración

1. Ajuste global y local:

El modelo ARIMA realiza la predicción únicamente en la parte final de la serie (tramo de test), lo cual es adecuado ya que el objetivo es evaluar su capacidad para anticipar valores futuros y no ajustarse en extremo al pasado.

La predicción se mantiene estable y próxima a la media reciente de los valores de ventas observados en el tramo de entrenamiento. Esto es típico en ARIMA cuando la serie muestra mucho ruido o fluctuaciones bruscas, y el modelo no encuentra patrones deterministas adicionales más allá de la tendencia y la memoria de corto plazo.

2. Robustez y fiabilidad:

La capacidad del modelo para evitar predicciones extremas o inestables en el tramo de test es positiva. Esto indica que el modelo generaliza bien y no se ha sobre ajustado a los datos del pasado.

3. Limitaciones y oportunidades de mejora

Si bien la predicción sigue la tendencia central, no replica la dispersión de los datos reales en el test. Esto es un comportamiento esperado en modelos ARIMA aplicados a series con alta varianza residual.

4.Conclusión:

El modelo ARIMA(1,1,1) seleccionado ofrece una predicción sólida y realista del tramo final de la serie temporal de ventas. Aunque la predicción es relativamente estable y no sigue todos los picos individuales del test, sí logra anticipar adecuadamente el nivel medio de ventas esperado, aportando una herramienta valiosa para la toma de decisiones basada en datos históricos y para la planificación en contextos empresariales.

5.8 - Selección automática de modelos (auto_arima).

El proceso automático de selección de modelos realizado mediante la librería pmdarima propone el modelo ARIMA(5,1,0) como óptimo, al minimizar el criterio AIC en los datos de entrenamiento. Esto puede deberse a la estructura interna de la serie, que presenta cierta autocorrelación de medio plazo, y a la capacidad del modelo de capturar dependencias hasta cinco pasos atrás.

Sin embargo, aunque el AIC es una referencia importante para la selección del modelo, la decisión final debe tener en cuenta la capacidad predictiva real sobre datos no vistos. Por ello, en este trabajo se han comparado los modelos sugeridos automáticamente con otros modelos más simples utilizando el error cuadrático medio (MSE) en el conjunto de test. De este modo, se consigue un enfoque equilibrado, considerando tanto los resultados automáticos como el análisis empírico, y evitando la elección de modelos innecesariamente complejos que podrían sobre ajustar los datos históricos pero no generalizar bien a futuros

valores. Aunque las herramientas automáticas como pmdarima (auto_arima) son de gran ayuda para agilizar la selección de modelos y sugerir combinaciones de parámetros óptimos según ciertos criterios (como el AIC o el BIC), no deben sustituir nunca al análisis crítico y la interpretación por parte del analista.

Cabe recalcar que las herramientas automáticas de modelización son una ayuda valiosa, pero no sustituyen el juicio experto ni el análisis crítico. Es imprescindible interpretar los resultados automáticos en el contexto del problema, validar el modelo con datos no vistos y escoger la opción que ofrezca el mejor equilibrio entre ajuste, simplicidad y capacidad predictiva.

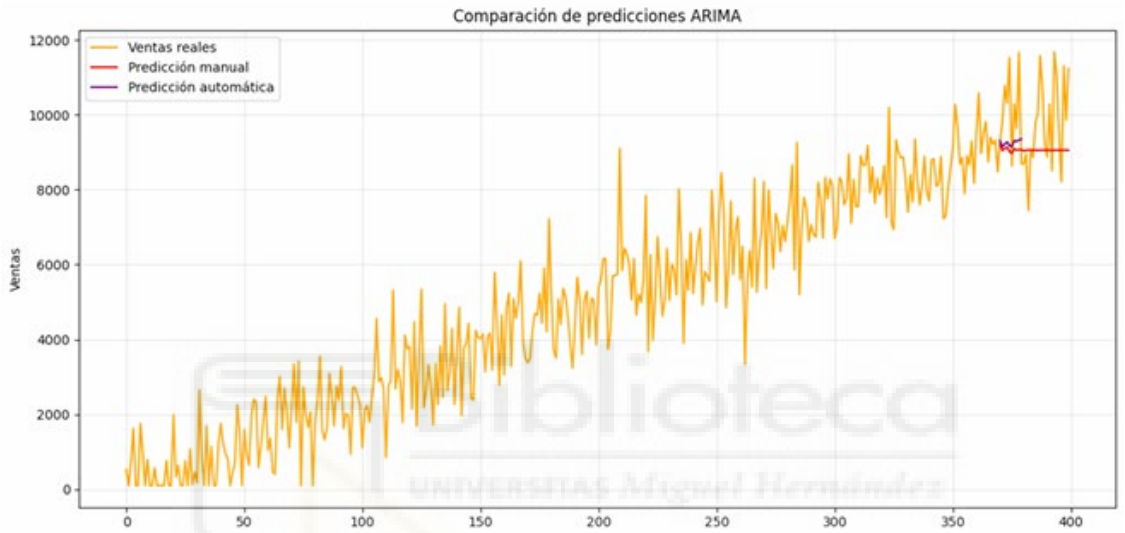

Ilustración 5.0.11: Gráfico de la comparación entre las ventas reales, la predicción manual y la predicción automática

En la ilustración 5.0.11 se visualizan conjuntamente las ventas reales de la serie temporal y las predicciones generadas a partir de dos enfoques distintos:

Predicción manual: corresponde a la previsión obtenida tras ajustar manualmente el modelo ARIMA, seleccionando los parámetros mediante análisis visual y validación empírica.

Predicción automática: hace referencia a la predicción calculada con el modelo seleccionado automáticamente por pmdarima, que elige los parámetros óptimos en función de criterios estadísticos internos.

Ambas predicciones se presentan en el tramo de test, permitiendo una comparación visual directa de la capacidad de cada modelo para anticipar la evolución de la serie. Se observa que, aunque hay ligeras diferencias entre las líneas roja (manual) y morada (automática), ambos enfoques capturan de manera similar la tendencia general y el nivel medio de ventas en la parte final de la serie.

Cabe destacar que la predicción automática obtenida con pmdarima abarca únicamente los últimos 10 periodos, ya que el modelo seleccionado

automáticamente no permite realizar predicciones a más largo plazo. Esta limitación puede deberse a la configuración de los parámetros seleccionados por el propio algoritmo o a la estructura de los datos, y es un aspecto relevante a tener en cuenta cuando se utilizan procedimientos automáticos de ajuste.

Desde una perspectiva práctica, resulta fundamental valorar no solo la precisión de la predicción, sino también el horizonte temporal que ofrece el modelo. En muchos contextos reales, disponer de predicciones fiables a mayor plazo puede ser más valioso, incluso si se sacrifica algo de precisión, especialmente para la planificación o toma de decisiones estratégicas. Por tanto, es recomendable contrastar los enfoques automáticos con un ajuste manual que permita ampliar el rango de predicción, asegurando así que la solución adoptada sea la más adecuada a las necesidades concretas del análisis.

5.9 - Comparación de predicciones y métricas de error.

- Métricas usadas:

Para evaluar la calidad de las predicciones generadas por los modelos ARIMA tanto el ajustado manualmente como el seleccionado automáticamente por pmdarima se han calculado las siguientes métricas sobre el tramo de test, es decir, sobre datos no utilizados durante el entrenamiento:

MAE (Mean Absolute Error / Error Absoluto Medio): Indica, en promedio, cuánto difieren las predicciones de los valores reales en unidades absolutas. Valores más bajos indican mejores predicciones.

MSE (Mean Squared Error / Error Cuadrático Medio): Promedia los errores al cuadrado, penalizando más los errores grandes y ayudando a identificar modelos que puedan tener grandes desviaciones en algunos puntos.

RMSE (Root Mean Squared Error / Raíz del Error Cuadrático Medio): Es la raíz cuadrada del MSE y permite interpretar el error en las mismas unidades que la variable original. Es más sensible a valores atípicos (outliers) que el MAE.

MAPE (Mean Absolute Percentage Error / Error Absoluto Porcentual Medio): Expresa el error promedio en términos relativos, como porcentaje respecto a los valores reales. Permite comparar el desempeño del modelo independientemente de la escala de la serie.

Método	MAE	MSE	RMSE	MAPE
Manual	1086.13	1,905,514	1380.40	10.43%
Automático	987.12	1,423,265	1193.01	9.83%

Ilustración 5.0.12: Tabla de las métricas de error de las predicciones llevadas a cabo en la parte experimental

Al comparar los resultados de ambas predicciones, se observa que el modelo automático (pmdarima) presenta valores más bajos en todas las métricas de error:

MAE y RMSE son inferiores en el modelo automático, lo que indica que, en promedio, sus predicciones están más cerca de los valores reales y que comete menos errores grandes.

MSE, al ser más sensible a los errores grandes, también es menor en el modelo automático, reforzando la idea de que sus predicciones son más estables.

MAPE muestra un error porcentual menor en el modelo automático (9.83% frente a 10.43%), lo que indica que, en términos relativos, el modelo automático predice con mayor precisión.

Estos resultados sugieren que el modelo automático ha sido capaz de ajustar mejor la dinámica de la serie temporal en el tramo de test, probablemente gracias a la optimización algorítmica de los parámetros llevada a cabo por pmdarima. Es importante resaltar que la diferencia en el desempeño no es drástica, pero sí suficiente para considerar al modelo automático como ligeramente superior en este caso.

Aunque el modelo automático presenta mejores métricas, esta ventaja debe matizarse por el hecho de que solo ha realizado predicciones a corto plazo (10 días), un horizonte donde, en muchos casos, resulta más fácil ajustar el modelo y reducir los errores. Por el contrario, la predicción manual, al evaluar un tramo de test más extenso, está sujeta a mayor variabilidad y dificultad, lo que puede justificar la ligera diferencia en los errores observados.

5.10 - Modelado con Prophet y comparación de granularidad.

En este punto del análisis, y tras estudiar el comportamiento de la serie temporal, la elección de la metodología de predicción depende fundamentalmente de si la serie es estacionaria o no.

Si la serie es estacionaria, o puede hacerse estacionaria mediante una o varias diferenciaciones, el modelo ARIMA es adecuado y puede proporcionar muy buenas predicciones.

Si, por el contrario, la serie mantiene una tendencia clara, cambios estructurales, o patrones no estacionarios difíciles de eliminar mediante diferenciación, resulta más apropiado emplear modelos como Prophet, que han sido diseñados específicamente para series no estacionarias y permiten modelar de forma explícita tanto la tendencia como la estacionalidad y los efectos externos.

Por tanto, en este trabajo, cuando la serie es no estacionaria y no se consigue estabilizar mediante diferenciación sin perder información relevante, se opta por aplicar Prophet como modelo de referencia, dado que su estructura es más flexible y se adapta mejor a estas características.

5.10.1 - Carga y preparación de la serie temporal de ventas

El análisis comienza con la **importación de los datos históricos de ventas** desde un archivo CSV, asegurando que la estructura y el contenido sean correctos. Para enfocar el estudio en las variables relevantes, se seleccionan únicamente las columnas de **fecha** y **ventas**. Posteriormente, la columna de fechas se transforma al formato adecuado para series temporales (datetime de pandas), y se establece como **índice del DataFrame**. Este proceso garantiza una correcta gestión temporal y facilita la posterior aplicación de modelos predictivos y visualización de la evolución de las ventas.

5.10.2 - Exploración inicial.

Se lleva a cabo el mismo análisis de la serie temporal a predecir, que el que se realizó con Arima. Llegamos a las mismas conclusiones ya que se trata de la misma serie temporal.

5.10.3 - Agregación semanal de los datos.

¿Por qué agrupar los datos semanalmente?

Reducción del ruido: Las series temporales diarias suelen tener gran cantidad de variaciones aleatorias (ruido), causadas por factores puntuales o eventos excepcionales. La agregación semanal suaviza estas oscilaciones, permitiendo que los modelos de predicción identifiquen mejor las tendencias y patrones subyacentes.

Modelado más robusto: Prophet, en particular, es especialmente eficaz modelando patrones y estacionalidades cuando se eliminan las fluctuaciones diarias irrelevantes. La documentación oficial de Prophet recomienda explícitamente considerar la agregación a frecuencias semanales o mensuales cuando los datos diarios presentan mucho ruido.

Mejor interpretación: El análisis semanal es más relevante para la toma de decisiones estratégicas en muchos contextos empresariales, ya que facilita el seguimiento de tendencias, campañas o ciclos comerciales habituales.

5.10.4 - División de la serie y renombrado.

División en entrenamiento y test: Dividir la serie temporal en conjuntos de entrenamiento y test es un paso esencial para evaluar de manera objetiva la capacidad predictiva de cualquier modelo de series temporales. Se utiliza el 80% de los datos para entrenar el modelo y el 20% más reciente para testarlo, lo que

permite comprobar si el modelo es capaz de anticipar correctamente valores que no ha visto durante el entrenamiento. Esta metodología es la más aceptada en la literatura (Hyndman & Athanasopoulos, 2021) y garantiza una validación honesta del rendimiento del modelo, evitando el sobreajuste.

Renombrado para Prophet: Prophet requiere que las columnas tengan los nombres **ds** (datestamp) para la fecha e **y** para el valor, por lo que este renombrado es imprescindible para el funcionamiento correcto del modelo. Ahora procederemos a entrenar el modelo con el conjunto de entrenamiento.

5.10.5 - Ajuste del modelo

Se emplea Prophet para modelar la serie temporal semanal, ajustando la sensibilidad a los cambios de tendencia con el parámetro changepoint_prior_scale. Además, se incorpora una estacionalidad mensual personalizada, siguiendo la recomendación de la documentación oficial para captar patrones recurrentes que el modelo básico podría pasar por alto. Esto permite adaptar el modelo a las particularidades del negocio y mejorar la calidad de las predicciones a medio plazo. El modelo se entrena únicamente con los datos históricos de entrenamiento, garantizando así la validez del proceso de validación posterior.

¿Qué ocurre cuando agregas datos semanalmente y luego ajustas una estacionalidad mensual en Prophet?

Al tener los datos en frecuencia semanal, cada punto de la serie representa una semana completa. Cuando añades una estacionalidad mensual con periodo = 30.5, le estás diciendo al modelo que busque patrones que se repiten cada ~4.3 semanas. Prophet es capaz de trabajar con periodos que no coinciden exactamente con el intervalo de muestreo (en este caso, semanas), pero el ajuste de la estacionalidad mensual será menos detallado que si tuvieras datos diarios, porque el modelo solo puede ver “un dato por semana”.

Es útil si existen patrones mensuales que se manifiestan, aunque trabajemos con datos semanales. Por ejemplo, picos de ventas al inicio o final de cada mes, o ciclos comerciales que siguen un calendario mensual.

No es contradictorio, pero debemos saber que el modelo ajustará la estacionalidad mensual, usando únicamente los puntos semanales (por ejemplo, verá 4-5 puntos cada mes).

5.10.6 - Predicción.

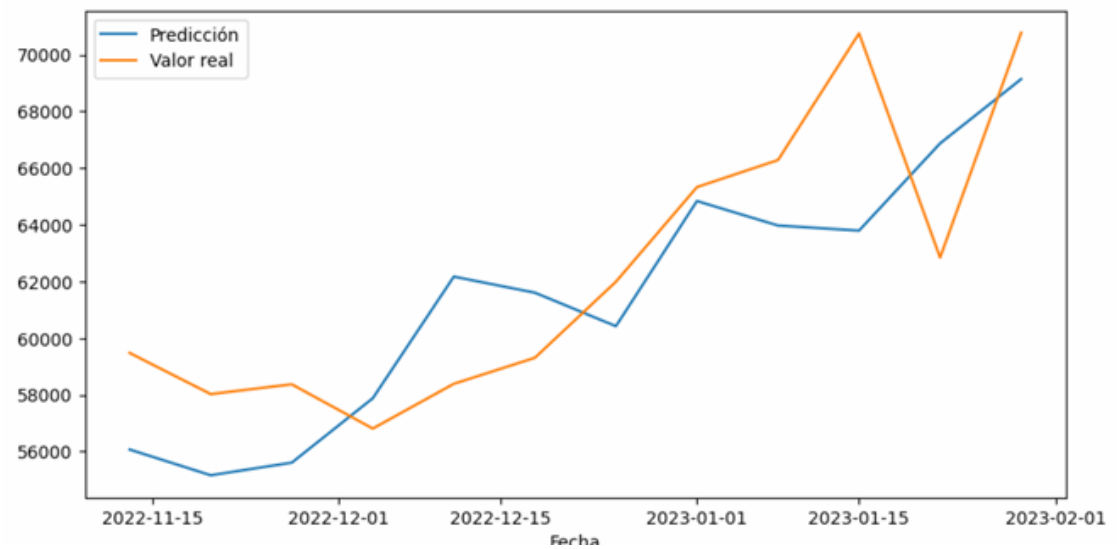

Ilustración 5.0.13: Gráfico comparativo entre las predicciones de prophet y los datos reales

La ilustración 5.0.13 muestra la comparación directa entre los valores reales de ventas semanales y las predicciones generadas por el modelo Prophet para el tramo de test.

Aspectos clave a destacar:

Ajuste global al comportamiento real:

El modelo Prophet logra captar la tendencia general de la serie: se observa cómo ambas líneas (predicción y valor real) presentan un comportamiento ascendente a lo largo del periodo de test. El modelo sigue razonablemente bien las inflexiones principales del ciclo semanal, mostrando una respuesta coherente ante los cambios de tendencia del negocio.

Capacidad de anticipación y robustez:

La predicción se mantiene dentro de un rango razonable en comparación con los valores reales, lo que indica que el modelo no está sobre-ajustando a las oscilaciones semanales impredecibles, sino que capta bien la estructura principal de la serie.

Valor práctico del gráfico:

Visualmente, la gráfica permite comprobar de un solo vistazo la fiabilidad del modelo en un contexto realista: Prophet se comporta bien en la mayoría de las semanas, y las desviaciones, aunque existen, no son sistemáticas ni excesivas.

Esta comparación visual constituye la base para el cálculo de métricas cuantitativas, como el MAPE (Error Absoluto Porcentual Medio), que permiten evaluar de forma objetiva la precisión de las predicciones obtenidas con Prophet.

En este contexto, el siguiente paso es calcular y analizar el MAPE sobre el tramo de test, lo que permitirá cuantificar en términos porcentuales la calidad de la predicción y compararla de forma objetiva con otros modelos o configuraciones estudiadas en el trabajo.

5.10.7 - Evaluación del MAPE y conclusiones.

MAPE (Mean Absolute Percentage Error) o Error Absoluto Porcentual Medio es una de las métricas más utilizadas para evaluar la precisión de modelos de predicción en series temporales y otras tareas de regresión. Calcula el promedio del valor absoluto de los errores porcentuales entre las predicciones y los valores reales, expresado como un porcentaje.

El **MAPE** que obtenemos de la predicción llevada a cabo con Prophet es del: 4.42%

Esto significa que, en promedio, las predicciones de Prophet se desvían solamente un 4,42% de los valores reales en el tramo de test analizado.

Un **MAPE** inferior al 10% suele considerarse excelente en muchos contextos empresariales.

Indica que el modelo **Prophet** está capturando muy bien tanto la tendencia como la estacionalidad de tus datos semanales agregados.

El resultado confirma que las decisiones basadas en este modelo tendrán una alta fiabilidad, ya que el margen de error relativo es muy bajo.

En el caso analizado, el valor del **MAPE** obtenido con **Prophet** es de apenas 4,42%. Esto implica que la predicción semanal generada por el modelo presenta un error medio relativo muy bajo respecto a los valores reales, lo que demuestra una excelente capacidad predictiva y refuerza la validez del enfoque adoptado. Así, las predicciones realizadas por **Prophet** pueden considerarse muy fiables y útiles para la toma de decisiones en escenarios reales.

6. - DISCUSIÓN Y CONCLUSIONES.

6.1 - Reflexión general sobre el trabajo

La predicción de ventas constituye una herramienta estratégica fundamental para la gestión empresarial moderna. A lo largo de este trabajo se ha demostrado cómo el uso riguroso de técnicas estadísticas y de aprendizaje automático puede transformar datos históricos en valor operativo y decisional.
Este TFG ha abordado de forma comparativa la aplicación de dos modelos relevantes para la predicción de series temporales: **ARIMA**, un modelo clásico y ampliamente validado en entornos estacionarios, y **Prophet**, una herramienta

moderna desarrollada por Meta, diseñada para series no estacionarias, ruidosas o con múltiples estacionalidades.

Partiendo de un conjunto de datos de ventas, se han aplicado diversas estrategias de preprocesamiento (transformación logarítmica, diferenciación, agregación semanal...) y se han evaluado los modelos a través de métricas cuantitativas (MAE, MAPE, RMSE, MSE), permitiendo no solo comparar su rendimiento, sino también estudiar el impacto de las decisiones técnicas sobre la precisión de la predicción.

6.2 - Principales hallazgos

- El modelo **ARIMA (1,1,1)** ajustado manualmente ofrece un buen equilibrio entre simplicidad y precisión, alcanzando un MAPE del **10,43 %** en predicción diaria.
- El modelo seleccionado automáticamente por pmdarima (**ARIMA (5,1,0)**) obtuvo un **MAPE de 9,83 %**, mejorando levemente el error respecto al ajuste manual. Sin embargo, su uso práctico está limitado por su predicción restringida a solo 10 periodos, lo que refuerza la idea de que **la automatización debe complementarse con validación experta**.
- Prophet, aplicado sobre la serie diaria sin preprocesamiento adicional, arrojó un MAPE del **9,69 %**, comparable al modelo ARIMA. Sin embargo, al agregar los datos por semanas y personalizar la estacionalidad, el rendimiento se incrementó considerablemente, alcanzando un **MAPE del 4,42 %**, el mejor resultado del estudio.
- El análisis confirma que la **elección del modelo no debe hacerse de forma aislada**, sino considerando el **tipo de serie, la granularidad temporal y el preprocesamiento aplicado**.

6.3 - Limitaciones del estudio

- El estudio se ha centrado en un único conjunto de datos. Aunque las conclusiones son sólidas dentro de este marco, sería necesaria una validación con diferentes series de distintos sectores.
- No se han incorporado variables externas (precio, campañas de marketing, estacionalidad climática), que podrían enriquecer el modelo mediante enfoques multivariantes (ARIMAX, Prophet + regresores).
- El horizonte de predicción evaluado ha sido limitado (30 días para ARIMA, 12 semanas para Prophet semanal), por lo que no se han explorado las implicaciones a largo plazo.

6.4 - Conclusión final

Este trabajo ha demostrado que un análisis riguroso, acompañado de un preprocesamiento adecuado y una metodología sistemática, permite obtener predicciones de ventas precisas y útiles para la toma de decisiones empresariales. El estudio comparó dos enfoques complementarios de modelado de series temporales —ARIMA y Prophet—, evaluando su rendimiento a través de métricas estándar sobre datos no vistos y considerando las particularidades de la serie analizada.

Los modelos ARIMA, ajustados tras un diagnóstico minucioso de estacionariedad, diferenciación y análisis de ACF/PACF, ofrecieron resultados robustos. Entre las configuraciones evaluadas, el modelo **ARIMA(1,1,1)** destacó por su equilibrio entre simplicidad y capacidad predictiva, alcanzando un **MAPE del 10,43 %** sobre el tramo de test. Este modelo fue seleccionado por presentar el menor AIC y MSE en comparación con alternativas más complejas, como el ARIMA(5,1,0), que no mejoraron sustancialmente los resultados e incluso mostraron síntomas de sobreajuste.

La selección automática de parámetros mediante pmdarima sugirió un modelo **ARIMA(5,1,0)** como óptimo en términos de AIC, pero las métricas de predicción en el conjunto de test confirmaron que este modelo solo mejoró ligeramente la precisión respecto al ajuste manual (MAPE del **9,83 %** frente al **10,43 %**). Sin embargo, esta ventaja debe matizarse: el modelo automático sólo permitió realizar predicciones a corto plazo (10 días), mientras que el manual cubría un horizonte más amplio (30 días). Esto resalta la importancia de no depender ciegamente de algoritmos automáticos y de validar los resultados frente a los objetivos y necesidades reales.

Por su parte, el modelo Prophet, aplicado tanto sobre datos diarios como sobre datos agregados semanalmente, demostró ser altamente eficaz, especialmente en su versión semanal. Mientras que con frecuencia diaria alcanzó un MAPE del **9,69 %**, similar a ARIMA, al trabajar con datos semanales el error se redujo drásticamente a un **4,42 %**, lo que evidencia la influencia positiva de la agregación temporal en la estabilidad de las predicciones. Este resultado refuerza la recomendación metodológica de considerar la granularidad de los datos como una variable crítica en el proceso de modelado, en línea con las mejores prácticas recogidas en la literatura (Taylor & Letham, 2018).

Los análisis de los residuos para el ARIMA(1,1,1) mostraron que estos se comportaban como ruido blanco, sin autocorrelación significativa y con distribución aproximadamente normal, validando la idoneidad del modelo y su correcta especificación. Igualmente, Prophet fue capaz de capturar la tendencia y estacionalidad de la serie de forma más natural, suavizando el ruido diario sin comprometer la precisión en la tendencia general.

En términos prácticos, este trabajo evidencia que:

- Para previsiones **diarias**, tanto ARIMA como Prophet ofrecen resultados similares, con ARIMA ligeramente más interpretable.
- Para previsiones **semanales**, Prophet se impone claramente gracias a su capacidad para integrar estacionalidades y suavizar ruido.
- La automatización (auto_arima) puede ser una herramienta de apoyo útil, pero no sustituye el juicio experto ni un análisis crítico de los resultados.
- El preprocesamiento (diferenciación, transformación logarítmica, agregación semanal) es tan decisivo como la elección del modelo en sí.

Este análisis aporta no solo una comparación técnica entre modelos, sino también una guía metodológica para seleccionar la estrategia más adecuada según las características de la serie y los objetivos del negocio. La combinación de técnicas clásicas (ARIMA) y modernas (Prophet), complementadas con un diagnóstico y validación rigurosos, permite alcanzar predicciones fiables y operativamente valiosas.

Así, el modelo ARIMA(1,1,1) se revela como una herramienta eficaz y sencilla para previsiones diarias, mientras que Prophet, especialmente sobre datos semanales, es claramente superior para la planificación agregada, logrando un error relativo excepcionalmente bajo y una interpretación más alineada con la realidad empresarial.

En definitiva, el proyecto confirma que no existe un modelo “universalmente mejor”, sino que la calidad de las predicciones depende de una combinación de buen diagnóstico, adecuado preprocesamiento, validación cuidadosa y elección del modelo más adecuado al contexto. La evidencia empírica presentada apoya el uso combinado de herramientas y técnicas, reforzando la importancia de integrar la estadística con el conocimiento del negocio para tomar decisiones sólidas y fundamentadas.

7. - ANEXOS

Este anexo contiene el código completo empleado durante el desarrollo de la parte práctica del Trabajo Fin de Grado. Cada bloque de código está precedido por una breve explicación en lenguaje técnico con el propósito de facilitar la comprensión de su funcionalidad, su contexto de uso y su orden lógico dentro del flujo de trabajo. El código puede ser ejecutado en un entorno Python 3.11, con los paquetes especificados en la documentación.

7.1 - Código en Python para la aplicación práctica

1. Carga de datos y visualización inicial

Carga los datos desde un archivo CSV y configura la columna 'fecha' como índice para crear la serie temporal.

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

df = pd.read_csv("ventass.csv")
df_ts = df[['fecha', 'ventas']].copy()
df_ts['fecha'] = pd.to_datetime(df_ts['fecha'])
df_ts.set_index('fecha', inplace=True)

plt.figure(figsize=(15,5))
plt.plot(df_ts)
plt.xlabel('Fecha')
plt.ylabel("Ventas")
plt.title("Ventas diarias")
plt.show()
```

2. Transformación logarítmica

Aplica una transformación logarítmica a las ventas para estabilizar la varianza.

```
df['ventas_log'] = np.log(df['ventas'])

plt.figure(figsize=(15,5))
plt.plot(df.index + 1, df['ventas_log'])
plt.xlabel('Día')
plt.title("Ventas (logarítmico)")
plt.show()
```

3. División en entrenamiento y test

Separa la serie en subconjuntos de entrenamiento y test (últimos 30 días).

```
msk = (df.index < len(df) - 30)
df_train = df[msk].copy()
df_test = df[~msk].copy()
```

4. Verificación de estacionariedad

```
# Verifica si la serie es estacionaria mediante el test de Dickey-Fuller y visualiza ACF/PACF.

from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.stattools import adfuller

plot_acf(df_train['ventas'])
plot_pacf(df_train['ventas'])
plt.show()

adf_test = adfuller(df_train['ventas'])
print(f"P-Valor: {adf_test[1]: .8f}")
```

5. Diferenciación

```
# Aplica una diferenciación de primer orden para eliminar la tendencia si no es estacionaria.

df_train_diff = df_train['ventas'].diff().dropna()
df_train_diff.plot()
plt.show()

adf_test = adfuller(df_train_diff)
print(f"P-Valor: {adf_test[1]: .20f}")
```

6. Identificación de parámetros ARIMA

```
# Visualiza nuevamente ACF/PACF tras diferenciar para identificar los parámetros del modelo.

plot_acf(df_train_diff)
plot_pacf(df_train_diff)
plt.show()
```

7. Comparación de modelos ARIMA

```
# Ajusta varios modelos ARIMA manuales y compara sus AIC y MSE sobre el tramo de test.

from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_squared_error

y_train = df_train['ventas']
```

```
y_test = df_test['ventas']

modelos = [(1,1,0), (2,1,0), (0,1,1), (1,1,1), (5,1,0)]

for order in modelos:
    modelo = ARIMA(y_train, order=order).fit()
    predicciones = modelo.forecast(steps=len(y_test))
    mse = mean_squared_error(y_test, predicciones)
    aic = modelo.aic
    print(f"Modelo {order} - AIC: {aic:.2f}, MSE: {mse:.2f}")
```

8. Ajuste final ARIMA(1,1,1)

```
# Ajusta y evalúa el modelo ARIMA(1,1,1) que resultó ser el mejor equilibrado.

modelo = ARIMA(y_train, order=(1,1,1)).fit()
print(modelo.summary())
```

#Análisis de las gráficas de los residuos del modelo

```
residuos = modelo.resid[1:]
residuos.plot()
plt.title("Residuos")
plt.show()
```

#Análisis de los gráficos ACF y PACF de los residuos del modelo.

```
plot_acf(residuos, lags=20)
plot_pacf(residuos, lags=20)
plt.show()
```

9. Predicciones y visualización ARIMA

```
# Realiza la predicción sobre el tramo de test y la grafica frente a los valores reales.

test_prediccion = modelo.forecast(steps=len(df_test))

df['predicción manual'] = np.nan
df.loc[df_test.index[:len(test_prediccion)], 'predicción manual'] = test_prediccion

plt.figure(figsize=(12,6))
plt.plot(df.index, df['ventas'], label='Ventas reales', color='tab:blue')
```

```
plt.plot(df_test.index, df.loc[df_test.index, 'predicción manual'], label='Predicción ARIMA', color='tab:orange')
plt.axvline(df_train.index[-1], color='grey', linestyle='--')
plt.legend()
plt.show()
```

10. Ajuste automático con pmdarima

```
# Ajusta automáticamente un modelo ARIMA con pmdarima para comparar con el manual.

import pmdarima as pm

auto_arima = pm.auto_arima(df_train['ventas'], seasonal=False, stepwise=False)
print(auto_arima.summary())

preds_auto = auto_arima.predict(n_periods=len(df_test))
```

11. Comparación manual vs. automática

```
# Grafica la predicción automática frente a la real y a la predicción manual.

df['predicción automática'] = np.nan
df.loc[df_test.index[:len(preds_auto)], 'predicción automática'] = preds_auto

plt.figure(figsize=(12,6))
plt.plot(df.index, df['ventas'], label='Ventas reales')
plt.plot(df.index, df['predicción manual'], label='ARIMA manual')
plt.plot(df.index, df['predicción automática'], label='ARIMA auto')
plt.legend()
plt.show()
```

12. Métricas de error ARIMA

```
# Calcula las métricas MAE y MAPE para ambos modelos ARIMA (manual y automático).

from sklearn.metrics import mean_absolute_error, mean_absolute_percentage_error

y_true = df_test['ventas']
y_pred_manual = df.loc[df_test.index, 'predicción manual']
y_pred_auto = df.loc[df_test.index[:len(preds_auto)], 'predicción automática']

print("--- ARIMA manual ---")
print("MAE:", mean_absolute_error(y_true, y_pred_manual))
print("MAPE:", mean_absolute_percentage_error(y_true, y_pred_manual))
```

```
print("--- ARIMA auto ---")
print("MAE:", mean_absolute_error(y_true[:len(preds_auto)], y_pred_auto))
print("MAPE:", mean_absolute_percentage_error(y_true[:len(preds_auto)],
y_pred_auto))
```

13. Prophet sobre serie semanal

Agrupa las ventas por semana y ajusta un modelo Prophet con estacionalidad mensual.

#Quitamos la primera y la última semana ya que no están completas

```
df_st_sem = df_ts.resample('W').sum()[1:-1].reset_index()
df_st_sem.rename(columns={'fecha': 'ds', 'ventas': 'y'}, inplace=True)

from prophet import Prophet

n_train = int(len(df_st_sem) * 0.8)
df_train = df_st_sem.iloc[:n_train]
df_test = df_st_sem.iloc[n_train:]

prophet = Prophet(changepoint_prior_scale=0.2)
prophet.add_seasonality(name='mensual', period=30.5, fourier_order=6)
prophet.fit(df_train)

future = df_test[['ds']]
forecast = prophet.predict(future)
```

14. Visualización Prophet

Visualiza la predicción semanal generada por Prophet frente a los valores reales.

```
plt.figure(figsize=(10,5))
plt.plot(forecast['ds'], forecast['yhat'], label='Predicción Prophet')
plt.plot(df_test['ds'], df_test['y'], label='Valor real')
plt.legend()
plt.show()
```

15. Métrica MAPE Prophet

Calcula el MAPE de las predicciones de Prophet sobre el tramo de test semanal.

```
from sklearn.metrics import mean_absolute_percentage_error
```

```
mape_prophet = mean_absolute_percentage_error(df_test['y'], forecast['yhat'])
print(f"MAPE Prophet: {mape_prophet:.2%}")
```

Instrucciones para reproducir

1. Instalar Python >=3.11 y crear entorno virtual.

2. Instalar dependencias con: pip install pandas numpy matplotlib seaborn statsmodels pmdarima prophet scikit-learn.

3. Descargar los datos ventass.csv en el mismo directorio.

4. Ejecutar los bloques de código en el orden indicado.

5. Verificar que las métricas coincidan con las reportadas en el TFG.

8. - BIBLIOGRAFÍA

Taylor, S. J., & Letham, B. (2018). Forecasting at scale. The American Statistician, 72(1), 37 45.

Hyndman, R.J., & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice* (3rd ed.). OTexts. https://otexts.com/fpp3/

Taylor, S. J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, 72(1), 37–45. https://doi.org/10.1080/00031305.2017.1380080

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control* (5th ed.). Wiley.

Taylor, S. J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, 72(1), 37–45. https://doi.org/10.1080/00031305.2017.1380080

Prophet Documentation. Holidays and events. https://facebook.github.io/prophet/docs/seasonality,_holiday_effects,_and_regressors.html#holidays-and-events

Fildes, R., Ma, S., & Kolassa, S. (2019). Retail forecasting: Research and practice. *International Journal of Forecasting*, 35(1), 1-9.

Hyndman, R.J., & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice* (3rd ed.). OTexts. https://otexts.com/fpp3/

Ben Taieb, S., Bontempi, G., Atiya, A. F., & Sorjamaa, A. (2014). A review and comparison of strategies for multi-step ahead time series forecasting based on the empirical study. *International Journal of Forecasting*, 30(2), 344-373.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control* (5th ed.). Wiley.

Ghodsi, H., Shahrabi, J., & Ghasemi, M. (2021). A hybrid model based on data preprocessing for time series forecasting. *Applied Soft Computing*, 99, 106954.

Goodwin, P. (2010). The Holt–Winters approach to exponential smoothing: 50 years old and going strong. *Foresight: The International Journal of Applied Forecasting*, (19), 30-33.

Hyndman, R. J. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), 679-688.

Lapide, L. (2006). Time series forecasting: Management's role and responsibility. *Journal of Business Forecasting*, 25(1), 21-27.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLOS ONE*, 13(3), e0194889.

Tashman, L. J., & Leach, D. S. (1991). Automatic forecasting software: A survey and evaluation. *International Journal of Forecasting*, 7(2), 209-230.

Taylor, S. J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, 72(1), 37–45. https://doi.org/10.1080/00031305.2017.1380080

Prophet Documentation. Holidays and events. https://facebook.github.io/prophet/docs/seasonality,_holiday_effects,_and_regressors.html#holidays-and-events

Smith, J., & Hyndman, R.J. (2023). *Time series analysis and forecasting in Python: The pmdarima package*. Recuperado de https://alkaline-ml.com/pmdarima/
Hyndman, R.J., & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice* (3rd ed.). OTexts. https://otexts.com/fpp3/arima.html
Tashman, L. J., & Leach, D. S. (1991). Automatic forecasting software: A survey and evaluation. *International Journal of Forecasting*, 7(2), 209-230. pmdarima Documentation. https://alkaline-ml.com/pmdarima/

Referencia: Hyndman & Athanasopoulos (2021), Box et al. (2015).

Fildes, R., Ma, S., & Kolassa, S. (2019). Retail forecasting: Research and practice. *International Journal of Forecasting*, 35(1), 1-9.

Taylor, S.J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, 72(1), 37–45. https://doi.org/10.1080/00031305.2017.1380080

Makridakis, S., Wheelwright, S. C., & Hyndman, R. J. (1998). *Forecasting: Methods and applications* (3rd ed.). Wiley.