

**Modelo predictivo de ventas usando aprendizaje automático (Machine Learning)
para pronosticar las ventas diarias de una empresa.**

Diego Ivan Baron Jaramillo

Asesor

Fernando Luis Carrascal Porras

Universidad Nacional Abierta y a Distancia – UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería – ECBTI

Especialización en Ciencia de Datos y Analítica

2024

Dedicatoria

A mis hijas, lo más maravilloso de mi vida, a mi compañera de vida, a mi padre, hermanos y familia, y en especial a mi tía, quien estará orgullosa por este nuevo logro.

Agradecimientos

Fernando Luis Carrascal Porras – director proyecto de grado

Resumen

El proyecto se direcciona en desarrollar un modelo de Machine Learning, usando la librería Facebook Prophet para realizar predicciones en el área de ventas de una organización comercial, teniendo en cuenta el impacto de las vacaciones y días feriados en un periodo laboral. Se implemento la metodología CRISPDM y posteriormente se realizó la comparación con modelos tradicionales para medir la precisión del algoritmo. Después de realizar lo mencionado anteriormente se obtienen métricas concluyentes, las cuáles indican una buena adaptación del algoritmo al modelo de datos con resultados que proporcionan información valiosa para la toma de decisiones al interior de la empresa.

Palabras clave: Machine learning, regresión, Facebook Prophet, predictivo, pronósticos

Abstract

The project is aimed at developing a Machine Learning model, using the Facebook Prophet library to make predictions in the sales area of a commercial organization, taking into account the impact of vacations and holidays in a working period. The CRISPDM methodology was implemented and then compared with traditional models to measure the accuracy of the algorithm. After doing the above, conclusive metrics are obtained, which indicate a good adaptation of the algorithm to the data model with results that provide valuable information for decision-making within the company.

Keywords: Machine learning, regression, Facebook Prophet, predictive, forecasting

Tabla de contenido

Glosario	1
Introducción	4
Objetivos	7
Objetivo General	7
Objetivos Específicos	7
Planteamiento del problema	8
Justificación.....	11
Marco conceptual	13
Marco teórico	18
Metodología	21
Objetivo específico 1.....	24
Objetivo específico 2.....	32
Objetivo específico 3.....	41
Objetivo específico 4.....	46
Conclusiones	50
Bibliografía	51
Apéndices	54

Lista de Tablas

Tabla 1 <i>Comparativo Prophet con otros modelos</i>	6
Tabla 2 <i>Comparaciones métricas de los cuatro modelos</i>	46

Lista de Figuras

Figura 1 <i>Circuito para realizar pronósticos a escala</i>	14
Figura 2 <i>Librerías y bibliotecas a usar</i>	25
Figura 3 <i>Carga y asignación del dataset a una variable</i>	26
Figura 4 <i>Se crea la variable df y se le asigna el dataset con index reseteado</i>	26
Figura 5 <i>Características del dataset</i>	27
Figura 6 <i>Estadísticas descriptivas del dataset</i>	27
Figura 7 <i>Mapa de calor para detectar datos nulos</i>	28
Figura 8 <i>Histogramas de los campos más representativos</i>	28
Figura 9 <i>Comportamiento de las ventas entre enero de 2013 y agosto de 2015</i>	29
Figura 10 <i>Creación de campos año, mes y día.</i>	29
Figura 11 <i>Comportamiento de las ventas por día</i>	30
Figura 12 <i>Comportamiento de las ventas por mes</i>	30
Figura 13 <i>Comportamiento de las ventas por año</i>	31
Figura 14 <i>Cambio de nombre de las columnas Date y Sales</i>	32
Figura 15 <i>Transformación logarítmica y graficando los datos</i>	33
Figura 16 <i>Instanciación del modelo y entrenamiento el mismo</i>	33
Figura 17 <i>Creación del pronóstico a 30 días</i>	34
Figura 18 <i>Ejecución del método predict</i>	34
Figura 19 <i>Últimos registros del pronóstico</i>	35
Figura 20 <i>Grafica con los resultados del pronóstico</i>	35
Figura 21 <i>Comparativos datos reales vs pronostico</i>	36

Figura 22 <i>Métricas del modelo usando sklearn</i>	36
Figura 23 <i>Métricas del algoritmo prophet</i>	37
Figura 24 <i>Grafica del indicador MAPE</i>	37
Figura 25 <i>Creación del dataframe con días feriados y vacaciones</i>	38
Figura 26 <i>Modelo incluyendo el dataset de días feriados y vacaciones</i>	39
Figura 27 <i>Métricas del modelo usando sklearn</i>	39
Figura 28 <i>Métricas del algoritmo prophet</i>	40
Figura 29 <i>Implementación del algoritmo ARIMA</i>	41
Figura 30 <i>Estadísticas del modelo ARIMA</i>	42
Figura 31 <i>Se implementa en método AIC</i>	42
Figura 32 <i>Estadísticas con los puntos óptimos del modelo ARIMA</i>	43
Figura 33 <i>Métricas de rendimiento del modelo ARIMA</i>	43
Figura 34 <i>Implementación del algoritmo SARIMA</i>	44
Figura 35 <i>Estadísticas del modelo SARIMA</i>	44
Figura 36 <i>Métricas de rendimiento del modelo SARIMA</i>	45

Glosario

Pronóstico: El pronóstico es el proceso de estimación en situaciones de incertidumbre. El término predicción es similar, pero más general, y usualmente se refiere a la estimación de series temporales o datos instantáneos. El pronóstico ha evolucionado hacia la práctica del plan de demanda en el pronóstico diario de los negocios. La práctica del plan de demanda también se refiere al pronóstico de la cadena de suministros.

Por consiguiente, los pronósticos son procesos críticos y continuos que se necesitan para obtener buenos resultados durante la planificación, de un proyecto. Si los clasificamos respecto al tiempo que abarcan, se puede clasificar en:

Pronósticos a corto plazo: En las empresas modernas, este tipo de pronóstico se efectúa cada mes o menos, y su tiempo de planeación tiene vigencia de un año. Se utiliza para programas de abastecimiento, producción, asignación de mano de obra a las plantillas de trabajadores, y planificación de los departamentos de fabricación.

Pronósticos a mediano plazo: Abarca un lapso de seis meses a tres años. Este se utiliza para estimar planes de ventas, producción, flujos de efectivo y elaboración de presupuestos.

Pronósticos a largo plazo: Este tipo de pronóstico se utiliza en la planificación de nuevas inversiones, lanzamiento de nuevos productos y tendencias tecnológicas de materiales, procesos y productos, así como en la preparación de proyectos. El tiempo de duración es de tres años o más.

Estadística: La estadística se encarga de la recolección, acopio y análisis de información para optimizar los procesos de toma de decisiones, utiliza un conjunto de funciones matemáticas que describen su función y distribución de probabilidad cuyos parámetros no son desconocidos.

Machine Learning: El aprendizaje automático investiga cómo las computadoras y la inteligencia artificial permite a desarrollar técnicas para aprender o mejorar su rendimiento en función de los datos, y optimizar su proceso en el descubrimiento de nuevos patrones dado a su aprendizaje. (Han Jiawei, 2014)

Minería de Datos: La minería de datos es el proceso de descubrir patrones interesantes a partir de grandes cantidades de datos, como proceso de descubrimiento de información potencialmente útil y novedosa. Los patrones interesantes representan el conocimiento y las medidas de interés del patrón, ya sean objetivas o subjetivas, y se pueden usar para guiar el proceso de descubrimiento (Han Jiawei, 2014)

Procesamiento de datos: Consiste en seleccionar, filtrar, agregar, muestrear, limpiar o transformar los datos. Usualmente los datos en sus bases no se encuentran limpios, por lo que se hace necesario una preparación previa en tal forma que no se produzca interferencia por datos que no aportan a la búsqueda de conocimiento. Por ejemplo, para determinar la tasa de deserción escolar es probable que no se utilicen datos relacionados con el número de veces que se cambiaron los bombillos en la institución educativa. La depuración es un proceso que evita interferencias, re-procesos y datos que no aportan a responder preguntas centrales.

Predicción: Consiste en predecir el valor para un atributo específico de set de datos. Por ejemplo, en un modelo de predicción de tasas de aprobación escolar, predecir la probabilidad que la aprobación haya sido por fraude. La predicción también puede utilizarse para validar una hipótesis descubierta. (Aristizábal Fúquene, 2017)

Regresión: Consiste en realizar el análisis de la dependencia de algunos valores de atributo sobre los valores de otros atributos en la misma variable y la producción automática de un modelo que pueda predecir estos valores de atributos a nuevos registros. Por ejemplo, dados los resultados de calificación académico en un grupo de estudiantes, construir un modelo que permita predecir la probabilidad de calificación de un nuevo grupo. (Aristizábal Fúquene, 2017)

Visualización de modelos: Consiste en elaborar visualizaciones que hagan del descubrimiento de conocimiento un proceso de fácil comprensión e interpretación para los individuos. Estas visualizaciones pueden variar desde gráficas de dispersión e histogramas hasta coordenadas paralelas y películas en 3D. Estas visualizaciones sirven, por ejemplo, para analizar progreso académico por cohortes, realizar predicciones a través de gráficas de regresión lineal o algo tan simple como porcentajes en una gráfica de pastel. (Aristizábal Fúquene, 2017)

Análisis exploratorio de datos: Consiste en realizar una exploración interactiva de un conjunto de datos sin una gran dependencia en rasgos o modelos preconcebidos para así determinar patrones interesantes. Las representaciones gráficas se utilizan a menudo para explotar el poder de la visión y la intuición humana. (Aristizábal Fúquene, 2017)

Introducción

En el entorno dinámico y competitivo del mercado actual, las empresas comerciales enfrentan el constante desafío de optimizar sus estrategias de negocio para maximizar las ventas y la rentabilidad. La predicción precisa de la demanda es un factor crucial para lograr este objetivo, ya que permite a las empresas tomar decisiones informadas sobre la producción, el inventario, la asignación de recursos y las campañas de marketing. En este contexto, los modelos de Machine Learning (ML) han surgido como herramientas poderosas para el análisis y la predicción de series de tiempo, incluyendo las ventas diarias en empresas comerciales. Imagine que es gerente de una tienda de productos de consumo y desea predecir la próxima demanda de productos para gestionar mejor la oferta. Un enfoque razonable de aprendizaje automático para este escenario es ejecutar algún análisis de series temporales que implique comprender, modelar y hacer predicciones basadas en puntos de datos secuenciales. (Dieckmann, 2024)

El presente proyecto de grado tiene como objetivo principal realizar una Evaluación Comparativa del Modelo de Machine Learning con Facebook Prophet para la Predicción de Ventas Diarias en una Empresa Comercial. Se pretende comparar el rendimiento del modelo Prophet, ampliamente utilizado para pronosticar series de tiempo, con el de un modelo de Machine Learning personalizado, diseñado específicamente para el análisis de datos de ventas de la empresa en cuestión. Los pronósticos de la empresa pueden basarse en tendencias económicas, datos de ventas anteriores y comparaciones en la industria. Las empresas ya establecidas pueden predecir fácilmente las ventas futuras, que se basan en datos comerciales pasados. Las nuevas empresas deben crear sus pronósticos basándose en información que no está lo suficientemente verificada, como la inteligencia competitiva y la investigación de mercados. La previsión de

ventas permite conocer la fuerza laboral, los recursos y el flujo de caja de la empresa. Los datos de ventas predictivos son cruciales para que las empresas obtengan capi al de inversión. (Emir Zunic, 2020).

Si bien los pronósticos precisos y confiables pueden conducir a enormes ahorros y reducciones de costos al facilitar una mejor planificación de la producción y el inventario, precios competitivos y una planificación oportuna de las promociones, se ha demostrado que las malas estimaciones de ventas son costosas en este ámbito, ya que es bien sabido que la escasez de bienes causa menores ganancias y puede llevar fácilmente a la insatisfacción del cliente. Además, no sólo el exceso de inventario puede obligar a la tienda a vender productos a precios más bajos o, peor aún, dar lugar a cancelaciones de inventario, sino que niveles de inventario superiores a los necesarios también aumentan los costos de almacenamiento. (Zunic, 2020).

Tabla 1*Comparativo Prophet con otros modelos*

Característica	Prophet	ARIMA	Redes Neuronales Recurrentes (RNNs)	XGBoost
Estacionalidad	Maneja estacionalidad fuerte de manera efectiva	Requiere identificar y modelar la estacionalidad explícitamente	Pueden modelar estacionalidad compleja y adaptativa	Puede manejar estacionalidad, pero requiere configuración adicional
Efectos de días festivos	Incorpora efectos de días festivos de manera predefinida	Requiere identificar y modelar los efectos de días festivos manualmente	Pueden modelar efectos de días festivos complejos y adaptables	Puede manejar efectos de días festivos, pero requiere configuración adicional
No linealidad	Maneja no linealidad a través de cambios de tendencia suaves	Limitado a modelos lineales	Pueden modelar no linealidad compleja y relaciones de dependencia a largo plazo	Puede manejar no linealidad compleja a través de árboles de decisión
Interpretabilidad	Modelo aditivo con componentes interpretables	Modelo de caja negra, menos interpretable	Modelos complejos, la interpretabilidad puede variar	Interpretabilidad depende del algoritmo y la configuración
Facilidad de uso	Implementación simple y biblioteca amigable para el usuario	Requiere conocimiento estadístico para la selección de parámetros y la interpretación de resultados	Implementación más compleja, requiere experiencia en redes neuronales	Implementación más compleja, requiere experiencia en aprendizaje conjunto

Fuente. Autoría propia

Objetivos

Objetivo General

Evaluar la efectividad del modelo de Machine Learning, implementado con el algoritmo Facebook Prophet, en la predicción de ventas diarias en una empresa del sector comercial.

Objetivos Específicos

- Analizar los datos recopilados de las ventas diarias de una empresa del sector comercial mediante la metodología CRISPDM.
- Implementar el modelo de Machine Learning utilizando el algoritmo Facebook Prophet para predecir las ventas diarias de la empresa durante un período determinado.
- Desarrollar modelos de comparación basados en métodos tradicionales de pronóstico, como el método de ARIMA, para evaluar su rendimiento frente al modelo de Facebook Prophet.
- Comparar la precisión y la exactitud de las predicciones generadas por cada modelo utilizando métricas como el error absoluto medio (MAE) y el error porcentual absoluto medio (MAPE).

Planteamiento del problema

Una serie de tiempo $\vec{X}_t \in \mathbb{R}^n$ donde t representa el tiempo y n es la dimensión de las variables deseadas, es una colección de observaciones realizadas secuencialmente a lo largo del tiempo, el problema de pronóstico es inferir el valor futuro \vec{X}_{t+h} basados en los valores actuales y pasados de x donde h es el tiempo de entrega (Li, 2023). En el ámbito empresarial actual, la capacidad de prever con precisión las tendencias de ventas se ha vuelto esencial para la toma de decisiones estratégicas. En este contexto, se propone el desarrollo de un modelo de Machine Learning que emplea la reconocida librería Facebook Prophet para realizar predicciones en el área de ventas de una organización. Hemos observado dos temas principales en la práctica de creación de previsiones de negocio. En primer lugar, las técnicas de previsión completamente automáticas pueden resultar difíciles, y a menudo son demasiado inflexibles para incorporar suposiciones o heurísticas útiles. Segundo los analistas responsables de las tareas de ciencia de datos en la organización suelen tener profunda experiencia en el dominio sobre los productos o servicios específicos que respaldan, pero a menudo no tienen formación en previsión de series temporales. (Taylor SJ, 2017)

El enfoque principal del proyecto consiste en capturar de manera efectiva las variaciones en el comportamiento de las ventas, teniendo en cuenta factores clave que influyen en la dinámica comercial. En este sentido, se destaca la importancia de considerar el impacto de las vacaciones y días feriados dentro de un año laboral, ya que estos eventos pueden tener efectos significativos en los patrones de consumo y, por ende, en las cifras de ventas. Los datos de ventas de productos minoristas contienen múltiples ciclos estacionales de diferentes duraciones. Por ejemplo, los datos de ventas diarias de cerveza mostrados en un experimento

muestran ciclos tanto semanales como anuales. Las ventas son altas durante los fines de semana y bajas durante los días laborables, altas en verano y bajas en invierno, y altas alrededor de Navidad. Algunos datos de ventas dependen de la naturaleza del negocio y de la ubicación del negocio. Por tanto, los modelos utilizados en la previsión deben controlar múltiples patrones estacionales. (Emir Zunic, 2020)

Las vacaciones y días feriados no solo representan períodos de posible aumento o disminución de la demanda, sino que también introducen variaciones en los hábitos de compra de los consumidores. Este fenómeno se convierte en un desafío para los métodos tradicionales de predicción de ventas, que a menudo no logran capturar adecuadamente estas fluctuaciones estacionales.

El uso de Facebook Prophet, una herramienta diseñada específicamente para abordar series temporales con patrones estacionales se presenta como una solución prometedora. Sin embargo, es esencial adaptar y entrenar el modelo de manera precisa para considerar las particularidades asociadas a las vacaciones y días feriados, maximizando así su capacidad predictiva. los métodos de previsión empresarial deben ser adecuado para 1) un gran número de personas que realizan pronósticos, posiblemente sin capacitación en métodos de series de tiempo; y 2) una gran variedad de problemas de pronóstico con potencial rasgos idiosincrásicos. (Taylor SJ, 2017)

En consecuencia, el presente proyecto busca llenar este vacío en la investigación y desarrollo de modelos de predicción de ventas, centrándose en la incorporación de factores estacionales relacionados con vacaciones y días feriados. La consecución de este objetivo no solo contribuirá a mejorar la exactitud de las predicciones, sino que también proporcionará a las organizaciones una herramienta valiosa para optimizar la gestión de inventarios, la planificación

de recursos y la toma de decisiones estratégicas en general. En el escenario del mundo real, el entorno empresarial en la industria minorista es muy dinámico y a menudo volátil, lo que se debe principalmente a los efectos de las vacaciones y al comportamiento de la competencia. Como resultado, a diferencia de los conjuntos de datos académicos ampliamente disponibles que se utilizan para demostrar y comparar varios métodos de pronóstico de series temporales, los datos de ventas del mundo real en este dominio conllevan varios desafíos, como datos históricos altamente no estacionarios, patrones de ventas irregulares y datos de ventas intermitentes. (Emir Zunic, 2020)

Los métodos tradicionales de análisis de series temporales, como ARIMA, a menudo presentan limitaciones que pueden afectar la precisión y la aplicabilidad de los pronósticos.

Algunas de estas limitaciones incluyen:

- Dificultad para manejar la estacionalidad compleja
- Falta de flexibilidad para modelar efectos de días festivos
- Incapacidad para capturar no linealidades
- Interpretabilidad limitada

Prophet aborda estas limitaciones de diversas maneras:

- Manejo efectivo de la estacionalidad
- Flexibilidad para modelar efectos de días festivos
- Capacidad para capturar no linealidades
- Interpretabilidad

Justificación

Las organizaciones del siglo XXI para ser más competitivas deben implementar y adaptar las nuevas tecnologías y conocimientos en cuanto a la ciencia de datos y el Machine Learning, y una de las áreas que pueden ayudar sustancialmente a las organizaciones son los pronósticos mediante las regresiones y análisis de series temporales, para esto existen muchos métodos y herramientas, una de ellas es Facebook ProPhet, que se caracteriza por usar series temporales con métodos optimizados y ajustados a las necesidades empresariales como se explica más adelante.

Antes de empezar a definir las diferentes técnicas vamos a hacer una breve introducción sobre el Machine learning. Como ya se ha comentado anteriormente, el ML es un método de análisis de datos que automatiza la construcción de modelos analíticos. Es una rama de la inteligencia artificial basada en la idea de que los sistemas pueden aprender de los datos, identificar patrones y tomar decisiones con la mínima intervención humana. Estos métodos nacieron de su aplicación al reconocimiento de patrones y de la idea de que las máquinas pueden aprender sin estar programadas para realizar tareas específicas. La aplicación de estas técnicas en diferentes disciplinas científicas surge del interés de algunos investigadores por ver si las máquinas podrían aprender de los datos que estos manejaban en sus investigaciones (Noemi, 2018).

Amazon utiliza algoritmos de Machine Learning para analizar el historial de compras, las búsquedas y la actividad en el sitio web de cada usuario para generar recomendaciones de productos personalizadas. Esto ha llevado a un aumento significativo en las ventas cruzadas y en el tamaño de los pedidos. (Romanchuk, 2023)

Netflix utiliza un sistema de recomendación basado en Machine Learning para sugerir películas y series de televisión a cada usuario en función de sus preferencias y hábitos de

visualización. Esto ha contribuido a aumentar la satisfacción del cliente y a reducir la tasa de cancelación. (Romanchuk, 2023)

Uber utiliza Machine Learning para predecir la demanda de viajes en diferentes áreas y momentos del día. Esto le permite asignar conductores de manera más eficiente y reducir los tiempos de espera para los pasajeros. (Romanchuk, 2023)

Marco conceptual

La previsión es una tarea común de la ciencia de datos que ayuda a las organizaciones con la planificación de la capacidad, el establecimiento de objetivos y la detección de anomalías. A pesar de su importancia, existen serios desafíos asociados con la producción de pronósticos confiables y de alta calidad, especialmente cuando hay una variedad de series temporales y los analistas con experiencia en modelado de series temporales son relativamente raros. Para abordar estos desafíos, describimos un enfoque práctico para el pronóstico "a escala" que combina modelos configurables con análisis de desempeño de analistas en el circuito. Proponemos un modelo de regresión modular con parámetros interpretables que pueden ser ajustados intuitivamente por analistas con conocimiento de dominio sobre la serie temporal. Describimos análisis de desempeño para comparar y evaluar procedimientos de pronóstico, y marcamos pronósticos automáticamente para revisión y ajuste manual. Las herramientas que ayudan a los analistas a utilizar su experiencia de manera más efectiva permiten realizar pronósticos confiables y prácticos de series temporales de negocios. (Taylor SJ, 2017)

Para que las empresas se vuelvan competitivas y disparen su crecimiento, necesitan aprovechar la IA/ML para desarrollar modelos predictivos para pronosticar las ventas en el futuro.

Los modelos predictivos intentan pronosticar las ventas futuras basándose en datos históricos, al tiempo que tienen en cuenta los efectos de la estacionalidad, la demanda, las vacaciones, las promociones y la competencia.

En este proyecto de grado, simularemos el rol de científico de datos en el departamento de ventas y usaremos una base de datos de 1115 tiendas como modelo. El objetivo es predecir las ventas diarias futuras en función de las características.

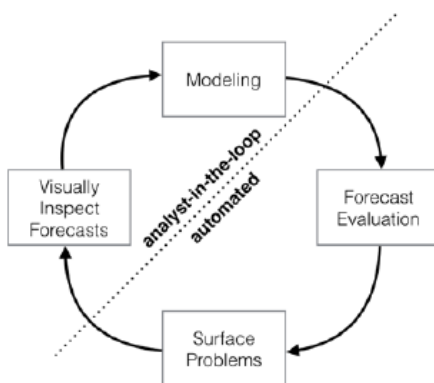
Facebook Prophet

Existe una amplia diversidad de problemas de previsión empresarial, sin embargo, existen algunas características, común a muchos de ellos. La figura 2 muestra una serie temporal representativa de Facebook. Los usuarios de Facebook pueden utilizar la plataforma de Eventos para crear páginas para eventos, invitar otros, e interactuar con los eventos de diversas maneras. La figura 2 muestra los datos diarios del número de eventos creados en Facebook. Hay varios efectos estacionales claramente visibles en esta época. serie: ciclos semanales y anuales, y una caída pronunciada alrededor de Navidad y Año Nuevo.

Ahora describimos un modelo de pronóstico de series temporales diseñado para manejar las características comunes de las series temporales de negocios que se ven en la Fig. 2. Es importante destacar que también está diseñado para tener unos parámetros que se pueden ajustar sin conocer los detalles del modelo subyacente.

Figura 1

Circuito para realizar pronósticos a escala



Fuente. (Taylor SJ, 2017)

Utilizamos un modelo de series temporales descomponibles (Harvey y Peters 1990) con tres componentes principales del modelo: tendencia, estacionalidad y días festivos. Se combinan en la siguiente ecuación: $y(t) = g(t) + s(t) + h(t) + \epsilon_t$

Aquí $g(t)$ es la función de tendencia que modela los cambios no periódicos en el valor de las series temporales, $s(t)$ representa cambios periódicos (por ejemplo, estacionalidad semanal y anual), y $h(t)$ representa los efectos de vacaciones que se produzcan en horarios potencialmente irregulares a lo largo de uno o más días. El término de error t representa cualquier cambio idiosincrásico que no sea acomodado por el modelo.

- Prophet es un software de código abierto lanzado por el equipo Core Data Science de Facebook.

- Prophet es un procedimiento para pronosticar datos de series de tiempo basado en un modelo aditivo donde las tendencias no lineales se ajustan a la estacionalidad anual, semanal y diaria, más los efectos de las vacaciones.

- Prophet funciona mejor con series de tiempo que tienen fuertes efectos estacionales y varias temporadas de datos históricos.

Prophet implementa un modelo de regresión aditiva con cuatro elementos:

- Prophet, lineal por partes, recoge automáticamente los puntos de cambio en los datos e identifica cualquier cambio en las tendencias.

- Un componente estacional anual modelado utilizando series de Fourier.

- Un componente estacional semanal.

- Una lista de vacaciones que se puede proporcionar manualmente.

Preciso Y Rápido

Los equipos de Facebook utilizan Prophet para realizar previsiones y planificaciones precisas. Prophet puede generar resultados en segundos.

Automático

No es necesario realizar un procesamiento previo de datos. Prophet trabaja con datos faltantes con varios valores atípicos.

Integración En El Campo De Conocimiento

Los usuarios pueden modificar el pronóstico agregando manualmente conocimientos específicos del dominio.

Aplicaciones campo energético. El modelo Prophet de Facebook y el modelo LSTM constituyen arquitecturas modernas de aprendizaje profundo que se pueden utilizar para establecer predicciones basadas en el consumo de carga promedio mediante el muestreo de secuencias similares en el pasado. En cierto modo, esto se parece a los modelos de persistencia N días y N mismos días introducidos en, donde la carga prevista para un intervalo de tiempo futuro se genera en función de la carga en intervalos de tiempo similares en días anteriores (Aleksei Kychkin, 2024).

Pronóstico de altas hospitalarias por vías respiratorias, se aplicaron cuatro enfoques de aprendizaje estadístico: RF, XGBOOST, PROPHET y el método ENSEMBLE a cada una de las siete regiones. Además, comparamos su capacidad predictiva tanto en el conjunto de

entrenamiento (2001-2018) como en el conjunto de prueba (2019) utilizando MSE, MAE e IS, así como MSE_{rel} e IS_{rel} para evaluar el sobreajuste (Shu Wei Chou-Chen, 2024).

Para estas últimas regiones, el método PROPHET constituye una alternativa interesante, ya que este método también permite incorporando un índice de tiempo que pueda explicar aquellos componentes no estacionarios que no sean explicados por las covariables (Shu Wei Chou-Chen, 2024).

Marco teórico

Es indudable que la virtualización del conocimiento ha desencadenado la apertura de este a todos los campos de la sociedad y en especial a las compañías mercantiles, así las cosas, desde la universidad se está transformando el modo de crear valor en las industrias. Los avances sobre las Tecnologías de la Información y Comunicación (TIC) están propiciando que las instituciones educativas organicen y construyan nuevos entornos virtuales para el aprendizaje y la enseñanza (Hernández Guerrero, 2018). De hecho, la educación a distancia (e-learning) es una forma de enseñanza en Internet que permite la creación de ambientes interactivos, eficientes y accesibles para el aprendizaje (Hernández Guerrero, 2018). Durante el Siglo XXI, los docentes están utilizando diversas aplicaciones web (p.ej., GeoGebra) bajo la modalidad e-learning con la finalidad de desarrollar las competencias en los estudiantes.

En los tiempos modernos se dice que el dato es un diamante en bruto de la era digital, ya que con este podemos realizar una transformación a la información, luego obtener unos análisis y posteriormente unos resultados, y estos últimos serán decisiones que se tomen luego de todo el proceso. La MD ya no se trata únicamente de un proceso de descubrimiento; es un sistema de apoyo analítico para la toma de decisiones ayudado por sistemas de información con unos objetivos mucho más específicos. Para lograr estos objetivos se emplean una serie de métodos que obedecen a algoritmos desarrollados en software específico para procesar la información de las bases de datos, métodos que tienen diferentes metas, técnicas y procesos dependiendo de lo que se busca en el proceso de Minería. Según (Calabria, 2014), las técnicas de la minería de datos resultan de combinar la inteligencia artificial y la estadística. Por tanto, no es sorprendente ver algoritmos relacionados con redes neuronales y otros con métodos de regresión estadísticos.

Un ejemplo práctico del uso de la ciencia de datos en la mejora de la productividad se dio en; determinar la producción potencial por unidad de superficie a partir de variables ecológicas, económicas o biológicas de fácil medición, de esta manera los resultados serán oportunos y económicos, aplicado a la resina de pino (Sánchez, 2013).

Otro ejemplo del uso de los modelos predictivos es; uso de técnicas automáticas de extracción de datos, para el análisis predictivo del tiempo que un graduado tarda en conseguir su primer empleo, orientadas a egresados de Ingeniería de Sistemas de la Universidad de Caldas, en el último lustro (Oscar Mauricio Bedoya Herrera, 2019).

Las aplicaciones de BlockChain usan modelos de pronóstico y el algoritmo de Facebook Prophet, como se muestra en el artículo de: “Comprenda el tiempo de espera en el mecanismo de tarifas de transacción: Una perspectiva interdisciplinaria” (Luyao Zhang, 2023).

Las grandes empresas minoristas como Walmart, Costco, Amazon, Target y otras tienen un modelo de negocio único en el que venden sus propios productos y los de la competencia en la misma tienda, ya sea en la tienda o en línea. En algunos casos, sus propios productos compiten con un producto de terceros. Estas empresas son un almacén de un volumen masivo de datos, que almacenan desde múltiples flujos (desde transacciones, eventos, inventario, por nombrar algunos). Empresas como Amazon y Walmart ayudan a los minoristas externos con soporte analítico a partir de los datos de series temporales que adquieren de cada producto y categoría, además de obtener información valiosa de los datos para maximizar sus ganancias comerciales. Tienen una enorme cantidad de transacciones de múltiples productos cada día. Analizar un volumen tan grande de datos y extraer información significativa siempre es un desafío. Sin embargo, con la llegada de la computación en la nube en los últimos años, el análisis de estos datos gigantescos en un período de tiempo reducido para resolver problemas comerciales, como

predecir las ventas futuras y la demanda futura, la recomendación de productos, etc., se han convertido en la clave del éxito de cualquier empresa (Md Rashidul Hasan, 2022).

Metodología

La metodología del proyecto se basará en las siguientes etapas:

Recolección y Preprocesamiento de Datos: Se procederá a la recopilación de datos históricos de ventas diarias de la empresa comercial, considerando un período de tiempo adecuado para el análisis. Los datos serán preprocesados para eliminar valores faltantes, inconsistencias y outliers.

Exploración y Análisis de Datos: Se realizará un análisis exploratorio de los datos de ventas para comprender su distribución, tendencias y patrones estacionales. Se identificarán factores externos que puedan influir en las ventas, como días festivos, eventos especiales o campañas de marketing.

Selección y Entrenamiento de Modelos: Se seleccionará un modelo de Machine Learning adecuado para la predicción de series de tiempo, considerando características como la precisión, la interpretabilidad y la complejidad computacional. Se entrenará el modelo seleccionado utilizando los datos preprocesados.

Evaluación del Modelo Prophet: Se implementará el modelo Prophet para la predicción de ventas diarias y se evaluará su rendimiento mediante métricas como el error medio absoluto (MAE) y el error cuadrático medio (RMSE).

Comparación de Modelos: Se comparará el rendimiento del modelo de Machine Learning personalizado con el del modelo Prophet en términos de precisión, interpretabilidad y complejidad computacional.

Conclusiones y Recomendaciones: Se presentarán las conclusiones obtenidas a partir de la evaluación comparativa de los modelos. Se formularán recomendaciones para la implementación del modelo de predicción más adecuado en la empresa comercial.

CRISP-DM es una metodología de minería de datos ampliamente utilizada que proporciona un marco estructurado para el desarrollo de proyectos de minería de datos exitosos. Su enfoque paso a paso facilita la comprensión y aplicación de las técnicas de minería de datos, incluso para aquellos con poca experiencia previa en este campo. (Witten, 2009).

Limpieza de datos:

- **Eliminación de casos incompletos:** Se eliminan las observaciones que contienen valores faltantes. Esta técnica puede ser útil si la cantidad de datos faltantes es pequeña.
- **Imputación de valores faltantes:** Se reemplazan los valores faltantes por valores estimados.
- **Imputación basada en modelos:** Se utilizan modelos estadísticos o de Machine Learning para predecir los valores faltantes.

Normalización de datos:

- **Escalamiento de características:** Las características de las series temporales pueden tener diferentes escalas, lo que puede afectar el rendimiento de los modelos de Machine Learning. Se pueden aplicar técnicas de escalamiento, como

el escalamiento mínimo-máximo o el escalado estándar, para normalizar las características a una escala común.

- **Transformación de datos:** En algunos casos, puede ser necesario transformar los datos para mejorar su distribución o eliminar patrones no deseados. Por ejemplo, se puede aplicar una transformación logarítmica a los datos para normalizar una distribución sesgada.

Manejo de outliers:

- **Detección basada en reglas:** Se establecen reglas basadas en la distribución de los datos para identificar valores que se alejan demasiado de la media o la mediana.
- **Detección basada en distancia:** Se calcula la distancia de cada punto al resto de los datos y se identifican como outliers aquellos que superan un umbral predefinido.
- **Detección basada en modelos:** Se utilizan modelos estadísticos o de Machine Learning para identificar outliers. (Mehandzhiyski, 2023)

Objetivo específico 1

- 1) Analizar los datos recopilados de las ventas diarias de una empresa del sector comercial mediante la metodología CRISPDM.

Se analizar una base de datos con 781 registros de sus ventas diarias en un periodo comprendido desde el 2 de enero de 2013 hasta el 31 de julio de 2015, se pronosticarán ventas a 30 días, inicialmente sin tener en cuenta los días feriados y de vacaciones y posteriormente se incluyen estas variables para comparar los resultados.

Campos de la base de datos a analizar:

Sales: ventas diarias, esta es la variable objetivo

Customers: número de clientes de un día dado
Open: Booleano para indicar si la tienda estaba abierta o cerrada (0 = cerrada, 1 = abierta)

Promo: describe si la tienda tenía algún tipo de promoción ese día o no

StateHoliday: indica si el día era festivo o no (a vacaciones públicas, b = vacaciones de Pascua holiday, c = Navidades, 0= No era festivo)

SchoolHoliday: indica si (Store, Date) se ve afectado por el cierre de las escuelas públicas

StoreType: categoría que indica el tipo de tienda (a, b, c, d)

Assortment: a = básico, b = extra, c = extedido

Competition Distance (en metros): distancia a la tienda de la competencia más cercana

CompetitionOpenSince [Mes/Año]: fecha en que abrió la competencia

Promo2: Promo2 es una promoción continuada y consecutiva en algunas tiendas (0 = la tienda no participa, 1 = la tienda participa)

Promo2Since [Año/Semana]: fecha en la que la tienda empieza a participar en la Promo2

PromoInterval: describe los intervalos consecutivos donde la Promo2 empieza, indicando los meses en los que empieza la misma. P.e. "Feb, May, Aug,Nov" significa que cada nueva ronda de promoción empieza en Febrero, Mayo, Agosto, Noviembre de cualquier año de esa tienda

Análisis exploratorio de los datos

Importamos las librerías a usar

Figura 2

Librerías y bibliotecas a usar

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import datetime
from prophet import Prophet
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")
from prophet.diagnostics import cross_validation
from prophet.diagnostics import performance_metrics
from prophet.plot import plot_cross_validation_metric
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
sns.set()
```

Fuente. Autoría propia

A la variable *sales_df* le asignamos el *dataset*

Figura 3

Carga y asignación del dataset a una variable

```
sales_df = pd.read_excel('tienda.xlsx', index_col='Date', parse_dates=True)
```

Fuente. Autoría propia

Creamos una variable llamada *df* y le asignamos el dataset con index reseteado

Figura 4

Se crea la variable df y se le asigna el dataset con index reseteado

```
df = sales_df.reset_index()
```

```
df.head()
```

	Date	DayOfWeek	Sales	Customers	Promo	StateHoliday	SchoolHoliday	StoreType	Assortment	CompetitionDistance	CompetitionOpenSinceMonth	Competition
0	2015-07-31	5	5263	555	1	0	1	c	a	1270.0	9.0	
1	2015-07-30	4	5020	546	1	0	1	c	a	1270.0	9.0	
2	2015-07-29	3	4782	523	1	0	1	c	a	1270.0	9.0	
3	2015-07-28	2	5011	560	1	0	1	c	a	1270.0	9.0	
4	2015-07-27	1	6102	612	1	0	1	c	a	1270.0	9.0	

Fuente. Autoría propia

Mostramos la información de los campos, el número de registros, tipo de dato y si hay registros nulos.

Figura 5

Características del dataset.

```
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 781 entries, 0 to 780
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Date                  781 non-null   datetime64[ns]
1   DayOfWeek             781 non-null   int64
2   Sales                 781 non-null   int64
3   Customers             781 non-null   int64
4   Promo                781 non-null   int64
5   StateHoliday         781 non-null   int64
6   SchoolHoliday        781 non-null   int64
7   StoreType            781 non-null   object
8   Assortment           781 non-null   object
9   CompetitionDistance  781 non-null   float64
10  CompetitionOpenSinceMonth 781 non-null   float64
11  CompetitionOpenSinceYear 781 non-null   float64
12  Promo2               781 non-null   int64
13  Promo2SinceWeek      781 non-null   float64
14  Promo2SinceYear      781 non-null   float64
15  PromoInterval        781 non-null   int64
dtypes: datetime64[ns](1), float64(5), int64(8), object(2)
memory usage: 97.8+ KB
```

Fuente. Autoría propia

Mostramos estadísticas descriptivas del *dataset*

Figura 6

Estadísticas descriptivas del dataset

```
df.describe()
```

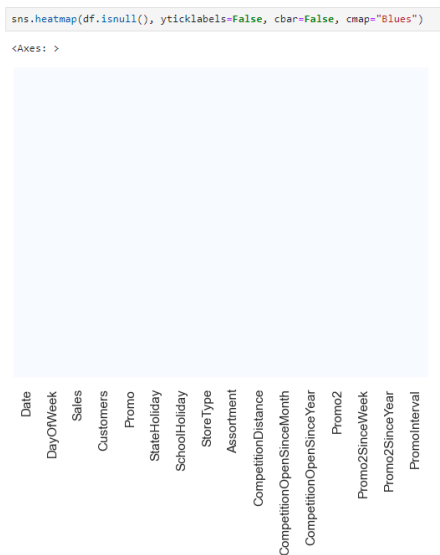
	Date	DayOfWeek	Sales	Customers	Promo	StateHoliday	SchoolHoliday	CompetitionDistance	CompetitionOpenSinceMonth	CompetitionOpenSinceYear
count	781	781.000000	781.000000	781.000000	781.000000	781.0	781.000000	781.0	781.0	781.0
mean	2014-04-16 02:38:33.956466176	3.504481	4759.096031	564.049936	0.448143	0.0	0.204866	1270.0	9.0	9.0
min	2013-01-02 00:00:00	1.000000	2362.000000	298.000000	0.000000	0.0	0.000000	1270.0	9.0	9.0
25%	2013-08-24 00:00:00	2.000000	4000.000000	500.000000	0.000000	0.0	0.000000	1270.0	9.0	9.0
50%	2014-04-14 00:00:00	3.000000	4647.000000	550.000000	0.000000	0.0	0.000000	1270.0	9.0	9.0
75%	2014-12-05 00:00:00	5.000000	5348.000000	612.000000	1.000000	0.0	0.000000	1270.0	9.0	9.0
max	2015-07-31 00:00:00	6.000000	9528.000000	1130.000000	1.000000	0.0	1.000000	1270.0	9.0	9.0
std	NaN	1.714656	1012.106393	93.707476	0.497622	0.0	0.403862	0.0	0.0	0.0

Fuente. Autoría propia

Mediante el un gráfico de mapa de calor determinamos si hay datos nulos en la base de datos, como se aprecia en la figura 7, no hay datos nulos en la base de datos.

Figura 7

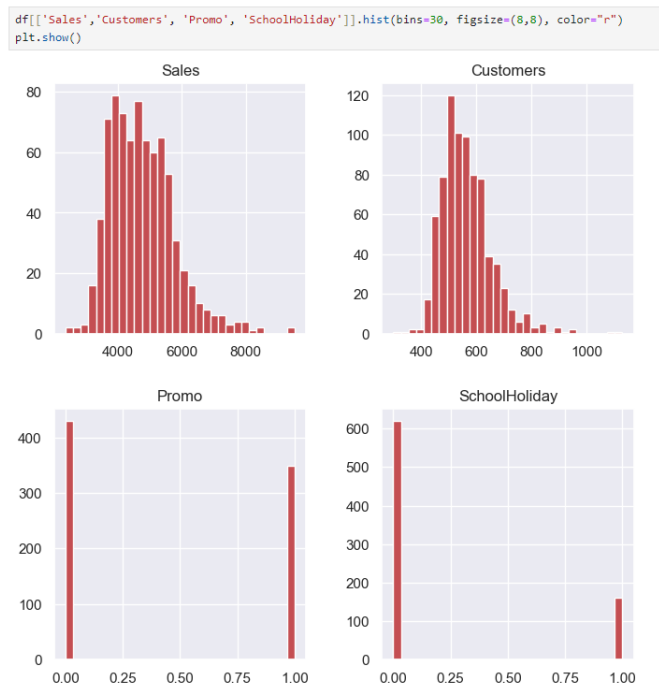
Mapa de calor para detectar datos nulos



Fuente. Autoría propia

Figura 8

Histogramas de los campos más representativos



Nota. El gráfico representa en que rango se concentra la mayor cantidad de datos según su variable, se aprecia que las ventas están entre 4000 y 6000 unidades monetarias

Graficamos el histórico de las ventas

Figura 9

Comportamiento de las ventas entre enero de 2013 y agosto de 2015

```
df.set_index('Date').Sales.plot(figsize=(13,5)).get_figure()
plt.title("Historico de Ventas")
plt.show()
```



Nota. Hay una tendencia a la baja, con algunos picos a finales de 2013 y comienzos de 2014. Así como a finales de 2014 e inicios de 2015.

Creamos las columnas año, mes y día para posteriormente graficar el comportamiento de los datos según estos campos.

Figura 10

Creación de campos año, mes y día.

```
df['Year'] = pd.DatetimeIndex(df['Date']).year
df['Month'] = pd.DatetimeIndex(df['Date']).month
df['Day'] = pd.DatetimeIndex(df['Date']).day
```

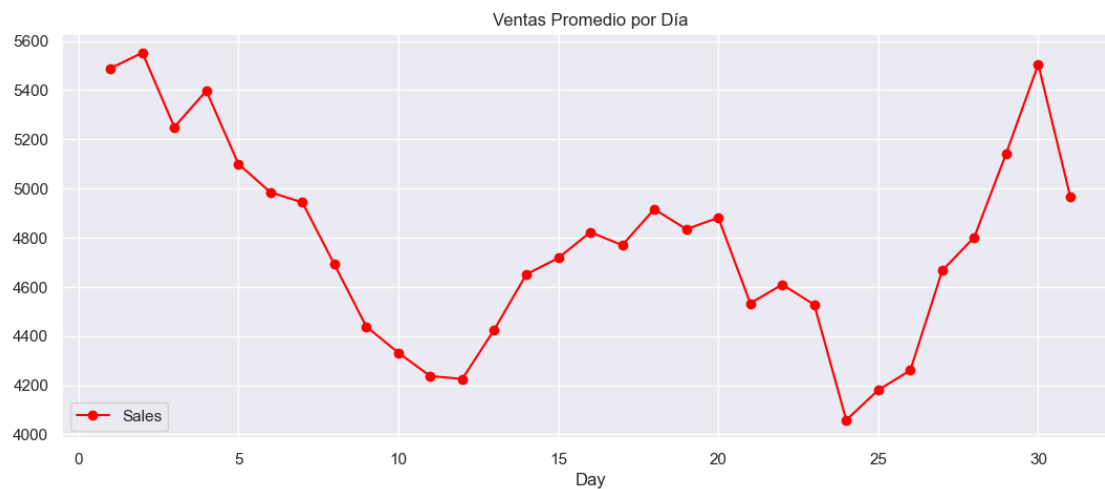
Nota. Creamos los campos año, mes y día a partir del campo fecha de la base de datos.

Graficamos el comportamiento promedio de las ventas por día.

Figura 11

Comportamiento de las ventas por día.

```
axis = df.groupby('Day')[['Sales']].mean().plot(figsize=(13,5), marker='o', color="red")
axis.set_title("Ventas Promedio por Día")
plt.show()
```

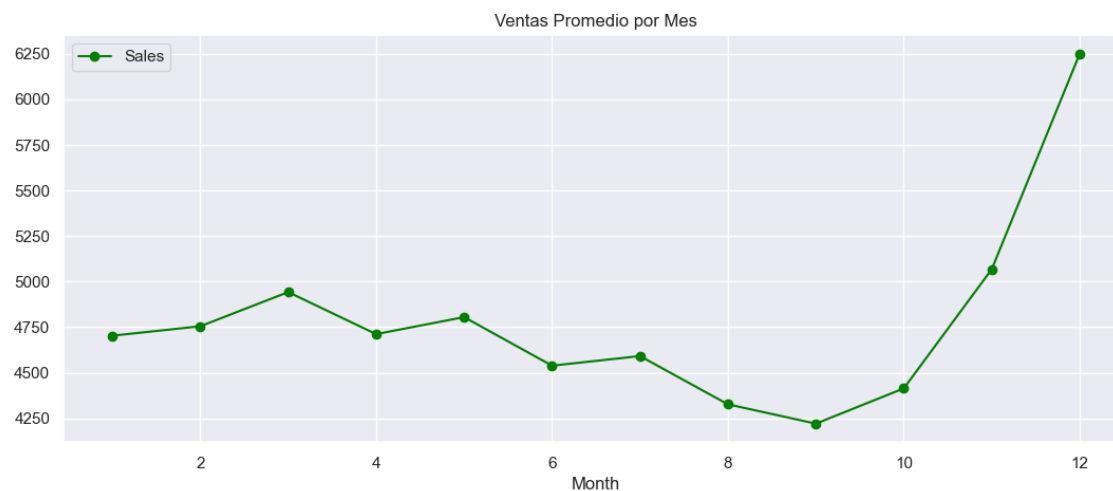


Nota. Se observa que los días donde más hay ventas son a inicios y finales del mes, días 1, 2 y 30.

Figura 12

Comportamiento de las ventas por mes.

```
axis = df.groupby('Month')[['Sales']].mean().plot(figsize=(13,5), marker='o', color="green")
axis.set_title("Ventas Promedio por Mes")
plt.show()
```



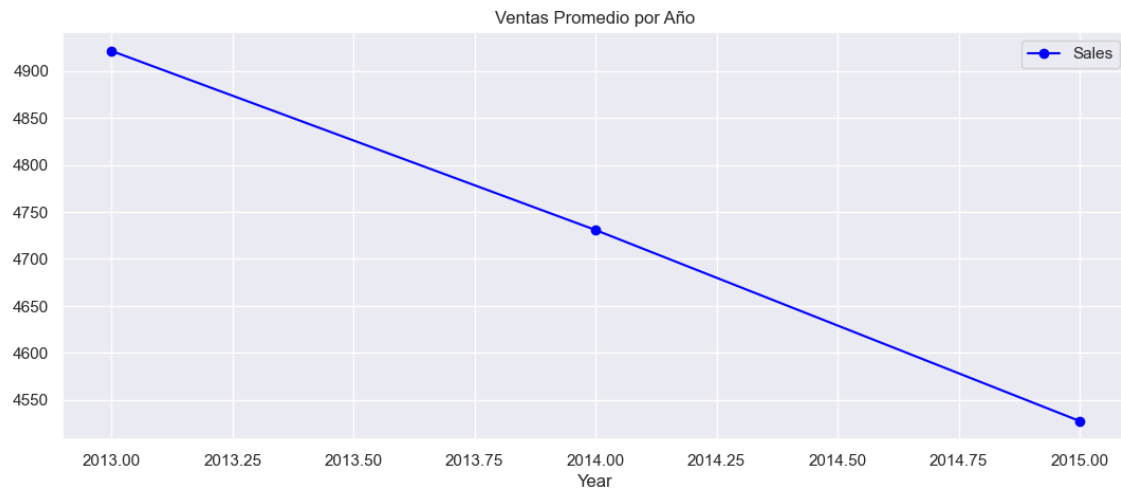
Nota. Los meses donde más se vende son noviembre y diciembre, esto puede ser por las festividades decembrinas.

Graficamos el comportamiento promedio de las ventas por año.

Figura 13

Comportamiento de las ventas por año.

```
axis = df.groupby('Year')[['Sales']].mean().plot(figsize=(13,5), marker='o', color="blue")  
axis.set_title("Ventas Promedio por Año")  
plt.show()
```



Nota. Se aprecia una caída leve de las ventas año a año.

Objetivo específico 2

2) Implementar el modelo de Machine Learning utilizando el algoritmo Facebook Prophet para predecir las ventas diarias de la empresa durante un período determinado.

El paso inicial para trabajar con la biblioteca de Facebook Prophet es renombrar las columnas de fecha y ventas (estos son los campos principales con los que se trabaja el algoritmo), como se muestra en la figura 14.

Figura 14

Cambio de nombre de las columnas Date y Sales

```
df=df.rename(columns={'Date':'ds', 'Sales':'y'})
df.head()
```

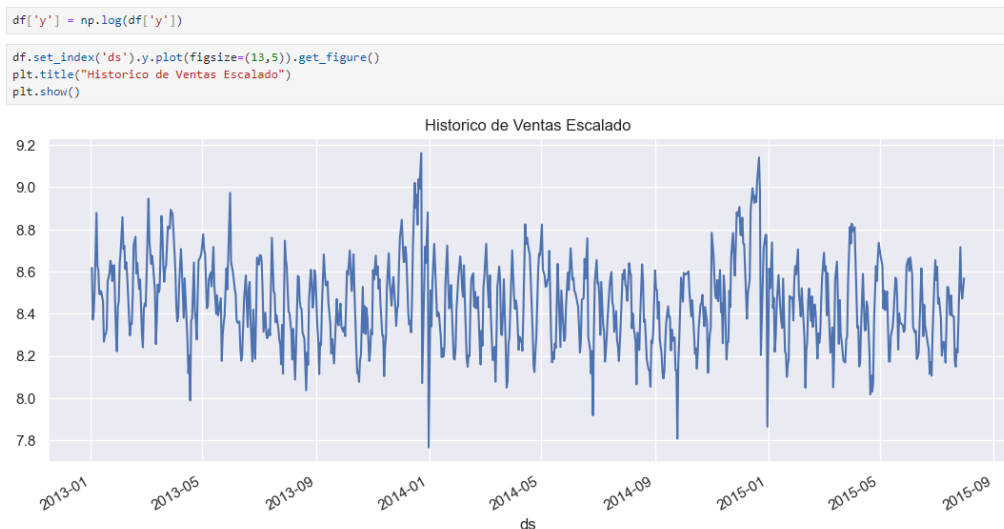
	ds	DayOfWeek	y	Customers	Promo	StateHoliday	SchoolHoliday	StoreType	Assortment	CompetitionDistance	CompetitionOpenSinceMonth
0	2015-07-31	5	8.568456	555	1	0	1	c	a	1270.0	9.0
1	2015-07-30	4	8.521185	546	1	0	1	c	a	1270.0	9.0
2	2015-07-29	3	8.472614	523	1	0	1	c	a	1270.0	9.0
3	2015-07-28	2	8.519391	560	1	0	1	c	a	1270.0	9.0
4	2015-07-27	1	8.716372	612	1	0	1	c	a	1270.0	9.0

Nota. Para poder implementar software de Facebook prophet, se debe renombrar las columnas que contienen la fecha y las ventas, fecha por “ds” y ventas por “y”

Cunado trabajamos con series temporales es recomendable echar un vistazo a los datos para determinar si existen anomalías (tendencia o datos estacionarios). Facebook Prophet utiliza la transformación logarítmica como una forma de eliminar alguna de estas anomalías, paso que se muestra en la figura 15.

Figura 15

Transformación logarítmica y graficando los datos



Nota. La grafica muestra el compartimiento de los datos transformados logarítmicamente donde se reducen o eliminan las anomalías de los datos.

Instanciamos el modelo de Facebook Prophet en la variable *model* y mediante el método *fit()* entrenamos el modelo.

Figura 16

Instanciación del modelo y entrenamiento el mismo.

```
model = Prophet()
model.fit(df)
```

```
17:48:28 - cmdstanpy - INFO - Chain [1] start processing
17:48:28 - cmdstanpy - INFO - Chain [1] done processing
```

```
<prophet.forecaster.Prophet at 0x2d1f7d84f80>
```

Nota. Se instancia la clase Prophet en la variable *model* y posteriormente se llama al método *fit*

Se realiza el pronóstico para treinta (30) días por lo cual agregamos este dato en el marco de trabajo que proporciona Facebook Prophet para esto e imprimimos los últimos registros.

Figura 17

Creación del pronóstico a 30 días.

```
future = model.make_future_dataframe(periods=30, freq = 'd')
future.tail()
```

```
ds
806 2015-08-26
807 2015-08-27
808 2015-08-28
809 2015-08-29
810 2015-08-30
```

Nota. De la clase model se llama al método `make_future_dataframe` y se le pasan los argumentos `periods=30`, `freq="d"`, lo que traduce que se quiere predecir a 30 días, esto se asigna a la variable `future`

Para pronosticar los datos futuros necesitamos ejecutar el modelo que proporciona Facebook Prophet. Y mostramos los últimos registros.

Figura 18

Ejecución del método `predict`.

```
forecast = model.predict(future)
```

```
forecast.tail()
```

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_terms_lower	additive_terms_upper	weekly	weekly_lower
806	2015-08-26	8.413995	8.069081	8.524475	8.413552	8.414337	-0.126464	-0.126464	-0.126464	-0.044398	-0.044398
807	2015-08-27	8.413896	8.057831	8.490748	8.413421	8.414278	-0.147404	-0.147404	-0.147404	-0.064908	-0.064908
808	2015-08-28	8.413797	8.099791	8.535844	8.413285	8.414205	-0.089117	-0.089117	-0.089117	-0.005655	-0.005655
809	2015-08-29	8.413698	8.148570	8.601815	8.413129	8.414125	-0.044494	-0.044494	-0.044494	0.040473	0.040473
810	2015-08-30	8.413599	8.115649	8.577469	8.412977	8.414054	-0.064832	-0.064832	-0.064832	0.022162	0.022162

Nota. A la variable `forecast` le asignamos el método `predict` de la clase `model` donde le pasamos el parámetro `future`, esto nos proporciona nuevos campos.

El modelo nos crea varias columnas, pero solo nos interesan las que componen la variable dependiente `y`.

Figura 19

Últimos registros del pronóstico.

```
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

	ds	yhat	yhat_lower	yhat_upper
806	2015-08-26	8.287532	8.069081	8.524475
807	2015-08-27	8.266492	8.057831	8.490748
808	2015-08-28	8.324680	8.099791	8.535844
809	2015-08-29	8.369205	8.148570	8.601815
810	2015-08-30	8.348767	8.115649	8.577469

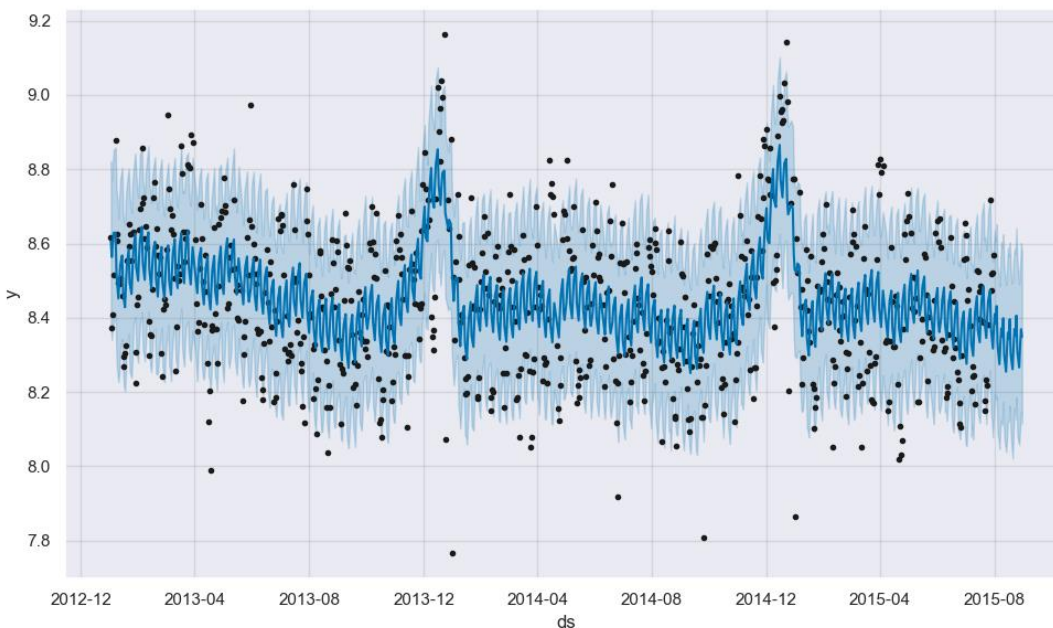
Nota. Se muestran los últimos cinco (5) registros del pronóstico con las columnas concernientes a la variable dependiente y.

Graficamos los resultados del modelo

Figura 20

Grafica con los resultados del pronóstico.

```
model.plot(forecast)
plt.show()
```

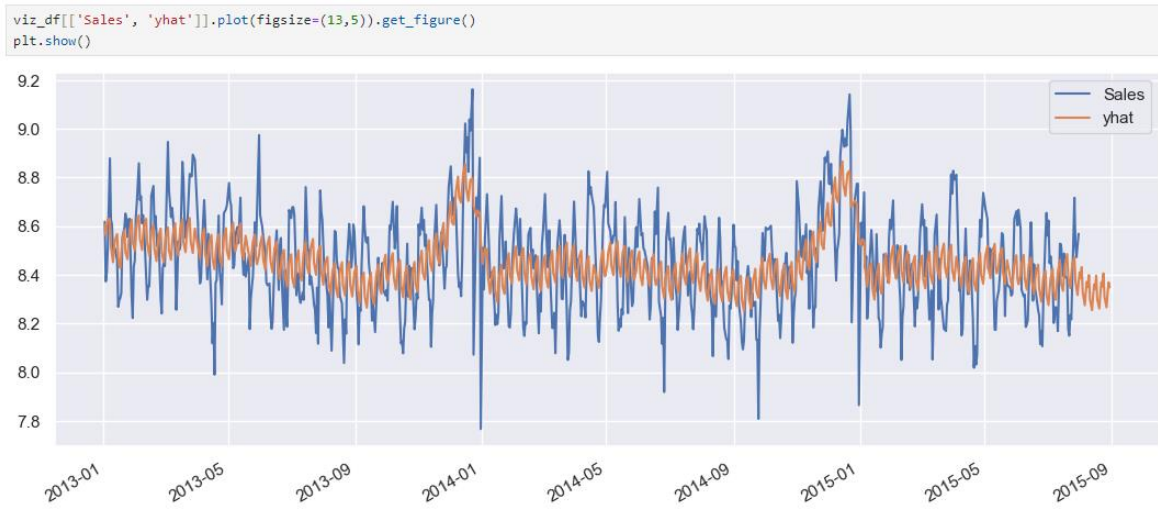


Nota. los puntos negros representan los datos originales, la línea azul representa el modelo y la sombra azul, los límites inferior y superior.

Ahora mostramos los datos originales y los del modelo

Figura 21

Comparativos datos reales vs pronostico.



Nota. Comparamos la curva de los datos iniciales contra la curva del pronóstico

Mostramos los resultados de las métricas que arroja *sklearn*

Figura 22

Métricas del modelo usando sklearn.

```
r2_score(metric_df.y, metric_df.yhat)
0.26162323820405275

mean_squared_error(metric_df.y, metric_df.yhat)
0.030949619863121537

mean_absolute_error(metric_df.y, metric_df.yhat)
0.144709609008503
```

Nota. Mediante las funciones de Sklearn mostramos las métricas de rendimiento del modelo, r^2 , error medio cuadrado y error absoluto.

Mostramos los resultados de las métricas que arroja Facebook Prophet

Figura 23

Métricas del algoritmo prophet.

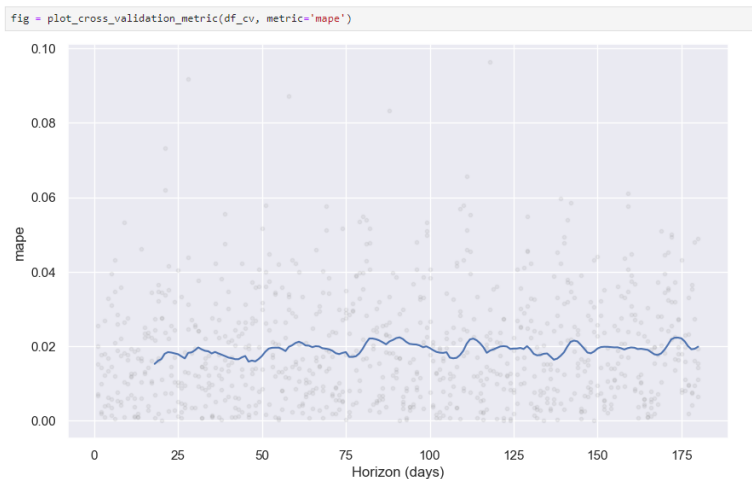
```
df_p = performance_metrics(df_cv)
df_p.head(n=13)
```

	horizon	mse	rmse	mae	mape	mdape	smape	coverage
0	18 days	0.025225	0.158825	0.128082	0.015252	0.013257	0.015147	0.826966
1	19 days	0.027645	0.166267	0.134789	0.016053	0.014565	0.015940	0.797753
2	20 days	0.028975	0.170220	0.138612	0.016524	0.016197	0.016398	0.773034
3	21 days	0.035742	0.189054	0.150162	0.017971	0.016415	0.017794	0.750562
4	22 days	0.037057	0.192501	0.153741	0.018409	0.016701	0.018225	0.741573
5	23 days	0.036231	0.190345	0.152746	0.018288	0.016701	0.018110	0.757303
6	24 days	0.035309	0.187906	0.151033	0.018063	0.016701	0.017904	0.766292
7	25 days	0.035379	0.188092	0.149400	0.017840	0.016226	0.017699	0.766292
8	26 days	0.034008	0.184414	0.144839	0.017261	0.014263	0.017145	0.773034
9	27 days	0.032294	0.179706	0.140586	0.016724	0.013129	0.016627	0.791011
10	28 days	0.040356	0.200887	0.153039	0.018213	0.014397	0.018097	0.764045
11	29 days	0.040866	0.202153	0.154261	0.018337	0.016026	0.018227	0.752809
12	30 days	0.042068	0.205106	0.159263	0.018904	0.016226	0.018806	0.752809

Nota. Facebook Prophet provee métodos para mostrar las métricas del rendimiento, mediante el método `performance_metrics`. Se muestra del día 18 hasta el día 30.

Figura 24

Grafica del indicador MAPE.



Nota. Mostramos el comportamiento del MAPE en un horizonte de 180 días.

Ahora vamos a incluir en el modelo los días feriados y de vacaciones y mostraremos nuevamente las métricas y observaremos si hay cambio en las mismas.

Figura 25

Creación del dataframe con días feriados y vacaciones.

```
# Obtenemos Las fechas de vacaciones
school_holiday = df[df['SchoolHoliday']==1].loc[:, 'ds'].values

# Obtenemos Las fechas de vacaciones publicas
state_holiday = df[(df['StateHoliday'] == 'a') | (df['StateHoliday'] == 'b') | (df['StateHoliday'] == 'c')].loc[:, 'ds'].values

# Creamos un DataFrame segun lo pide Prophet
school_holiday = pd.DataFrame({'ds': pd.to_datetime(school_holiday), 'holiday': 'school_holiday'})

state_holiday = pd.DataFrame({'ds': pd.to_datetime(state_holiday), 'holiday': 'state_holiday'})

# Concatenamos los dataset
festivos_vacaciones = pd.concat((state_holiday, school_holiday), axis=0)

festivos_vacaciones
```

	ds	holiday
0	2015-07-31	school_holiday
1	2015-07-30	school_holiday
2	2015-07-29	school_holiday
3	2015-07-28	school_holiday
4	2015-07-27	school_holiday
...
155	2013-01-07	school_holiday
156	2013-01-05	school_holiday
157	2013-01-04	school_holiday
158	2013-01-03	school_holiday
159	2013-01-02	school_holiday

Nota. Se crean dataset con los días feriados y vacaciones que se encuentran en la base de datos original

Incluimos el *dataset* al parámetro de días festivos y vacaciones.

Figura 26

Modelo incluyendo el dataset de días feriados y vacaciones.

```

: model = Prophet(holidays=festivos_vacaciones)
  model.fit(df)

21:36:16 - cmdstanpy - INFO - Chain [1] start processing
21:36:16 - cmdstanpy - INFO - Chain [1] done processing

: <prophet.forecaster.Prophet at 0x20ab5016090>

: future = model.make_future_dataframe( periods=30, freq = 'd' )
  future.tail()

:
   ds
806 2015-08-26
807 2015-08-27
808 2015-08-28
809 2015-08-29
810 2015-08-30

: forecast = model.predict( future )
  forecast.tail()

```

Nota. Corremos nuevamente el modelo incluyendo al argumento holidays el dataset de los días feriados y vacaciones.

Mostramos las métricas con este nuevo valor.

Figura 27

Métricas del modelo usando sklearn.

```
r2_score(metric_df.y, metric_df.yhat)
```

```
0.26566786310680657
```

```
mean_squared_error(metric_df.y, metric_df.yhat)
```

```
0.03078008635434118
```

```
mean_absolute_error(metric_df.y, metric_df.yhat)
```

```
0.14425377892607183
```

Nota. Mediante las funciones de Sklearn mostramos las métricas de rendimiento del modelo con el nuevo parámetro, r2, error medio cuadrado y error absoluto.

Hay una leve mejoría en cuanto a las meticas mediane sklearn.

Ahora revisemos las métricas con FaceBool Prophet.

Figura 28

Métricas del algoritmo prophet.

```
df_p = performance_metrics(df_cv)
df_p.head(n=13)
```

	horizon	mse	rmse	mae	mape	mdape	smape	coverage
0	18 days	0.024803	0.157490	0.126584	0.015062	0.013563	0.014966	0.838202
1	19 days	0.027083	0.164569	0.133009	0.015831	0.014912	0.015727	0.808989
2	20 days	0.028249	0.168075	0.136455	0.016258	0.015342	0.016140	0.784270
3	21 days	0.035033	0.187170	0.147878	0.017691	0.015392	0.017521	0.761798
4	22 days	0.036327	0.190598	0.151397	0.018121	0.015392	0.017945	0.752809
5	23 days	0.035508	0.188436	0.150413	0.018002	0.015392	0.017831	0.768539
6	24 days	0.034680	0.186226	0.148850	0.017796	0.015392	0.017642	0.777528
7	25 days	0.034805	0.186561	0.147325	0.017585	0.015342	0.017450	0.777528
8	26 days	0.033529	0.183109	0.143077	0.017044	0.013563	0.016933	0.784270
9	27 days	0.031905	0.178619	0.139276	0.016563	0.013363	0.016470	0.797753
10	28 days	0.040220	0.200548	0.152246	0.018113	0.014432	0.018001	0.764045
11	29 days	0.040757	0.201884	0.153619	0.018255	0.014912	0.018150	0.752809
12	30 days	0.042112	0.205211	0.159111	0.018881	0.016037	0.018787	0.730337

Nota. Facebook Prophet provee métodos para mostrar las métricas del rendimiento, incluyendo el parámetro de días feriados y vacaciones, mediante el método `performance_metrics`. Se muestra del día 18 hasta el día 30.

Hay una leve mejoría en algunas métricas.

Objetivo específico 3

3) Desarrollar modelos de comparación basados en métodos tradicionales de pronóstico, como el método de ARIMA, para evaluar su rendimiento frente al modelo de Facebook Prophet.

Desarrollamos el modelo en ARIMA según las indicaciones del libro (B V Vishwas, 2020).

Figura 29

Implementación del algoritmo ARIMA.

ARIMA

```

def timeseries_evaluation_metrics_func(y_true, y_pred):
    def mean_absolute_percentage_error(y_true, y_pred):
        y_true, y_pred = np.array(y_true), np.array(y_pred)
        return np.mean(np.abs((y_true - y_pred) / y_true)) * 100
    print('Evaluation metric results:-')
    print(f'MSE is : {metrics.mean_squared_error(y_true, y_pred)}')
    print(f'MAE is : {metrics.mean_absolute_error(y_true, y_pred)}')
    print(f'RMSE is : {np.sqrt(metrics.mean_squared_error(y_true, y_pred))}')
    print(f'MAPE is : {mean_absolute_percentage_error(y_true,y_pred)}')
    print(f'R2 is : {metrics.r2_score(y_true, y_pred)}',end='\n\n')

def Augmented_Dickey_Fuller_Test_func(series , column_name):
    print (f'Results of Dickey-Fuller Test for column: {column_name}')
    dftest = adfuller(series, autolag='AIC')
    dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','No Lags Used','Number of Observations Used'])
    for key,value in dftest[4].items():
        dfoutput['Critical Value (%s)%key] = value
    print (dfoutput)
    if dftest[1] <= 0.05:
        print("Conclusion:====>")
        print("Reject the null hypothesis")
        print("Data is stationary")
    else:
        print("Conclusion:====>")
        print("Fail to reject the null hypothesis")
        print("Data is non-stationary")

```

Nota. Se modela el dataset mediante el algoritmo ARIMA para posteriormente comparar las métricas.

Obtenemos los siguientes resultados:

Figura 30

Estadísticas del modelo ARIMA.

```
Augmented_Dickey_Fuller_Test_func(df['y' ],'Y')
Results of Dickey-Fuller Test for column: Y
Test Statistic      -3.755050
p-value            0.003402
No Lags Used       21.000000
Number of Observations Used  759.000000
Critical Value (1%) -3.438995
dtype: float64
Conclusion:====>
Reject the null hypothesis
Data is stationary
Test Statistic      -3.755050
p-value            0.003402
No Lags Used       21.000000
Number of Observations Used  759.000000
Critical Value (1%) -3.438995
Critical Value (5%) -2.865355
dtype: float64
Conclusion:====>
Reject the null hypothesis
Data is stationary
Test Statistic      -3.755050
p-value            0.003402
No Lags Used       21.000000
Number of Observations Used  759.000000
Critical Value (1%) -3.438995
Critical Value (5%) -2.865355
Critical Value (10%) -2.568802
dtype: float64
Conclusion:====>
Reject the null hypothesis
Data is stationary
```

Nota. Se muestran estadísticas después de correr el modelo ARIMA, como por ejemplo el valor p.

Figura 31

Se implementa en método AIC.

```
stepwise_model = auto_arima(train,start_p=1, start_q=1,max_p=7, max_q=7, seasonal=False,d=None, trace=True,
error_action='ignore',suppress_warnings=True,stepwise=True)
stepwise_model.summary()

Performing stepwise search to minimize aic
ARIMA(1,0,1)(0,0,0)[0] : AIC=-557.664, Time=0.17 sec
ARIMA(0,0,0)(0,0,0)[0] : AIC=5338.031, Time=0.02 sec
ARIMA(1,0,0)(0,0,0)[0] : AIC=inf, Time=0.13 sec
ARIMA(0,0,1)(0,0,0)[0] : AIC=4340.366, Time=0.06 sec
ARIMA(2,0,1)(0,0,0)[0] : AIC=-657.596, Time=0.92 sec
ARIMA(2,0,0)(0,0,0)[0] : AIC=inf, Time=0.12 sec
ARIMA(3,0,1)(0,0,0)[0] : AIC=-570.091, Time=0.36 sec
ARIMA(2,0,2)(0,0,0)[0] : AIC=-571.879, Time=1.07 sec
ARIMA(1,0,2)(0,0,0)[0] : AIC=-558.033, Time=0.25 sec
ARIMA(3,0,0)(0,0,0)[0] : AIC=inf, Time=0.15 sec
ARIMA(3,0,2)(0,0,0)[0] : AIC=-575.365, Time=1.41 sec
ARIMA(2,0,1)(0,0,0)[0] intercept : AIC=-695.725, Time=1.05 sec
ARIMA(1,0,1)(0,0,0)[0] intercept : AIC=-673.432, Time=1.31 sec
ARIMA(2,0,0)(0,0,0)[0] intercept : AIC=-673.549, Time=0.15 sec
ARIMA(3,0,1)(0,0,0)[0] intercept : AIC=-697.383, Time=1.55 sec
ARIMA(3,0,0)(0,0,0)[0] intercept : AIC=-688.615, Time=0.61 sec
ARIMA(4,0,1)(0,0,0)[0] intercept : AIC=-702.910, Time=1.49 sec
ARIMA(4,0,0)(0,0,0)[0] intercept : AIC=-690.244, Time=0.63 sec
ARIMA(5,0,1)(0,0,0)[0] intercept : AIC=-732.181, Time=2.73 sec
ARIMA(5,0,0)(0,0,0)[0] intercept : AIC=-732.730, Time=1.53 sec
ARIMA(6,0,0)(0,0,0)[0] intercept : AIC=-733.291, Time=1.30 sec
ARIMA(7,0,0)(0,0,0)[0] intercept : AIC=-732.743, Time=0.67 sec
ARIMA(6,0,1)(0,0,0)[0] intercept : AIC=-730.924, Time=1.31 sec
ARIMA(7,0,1)(0,0,0)[0] intercept : AIC=-730.412, Time=1.69 sec
ARIMA(6,0,0)(0,0,0)[0] : AIC=inf, Time=0.15 sec

Best model: ARIMA(6,0,0)(0,0,0)[0] intercept
Total fit time: 21.681 seconds
```

Nota. Mediante AIC se obtienen los puntos de intercepción óptimos del modelo.

Figura 32

Estadísticas con los puntos óptimos del modelo ARIMA.

[81]: SARIMAX Results

Dep. Variable:	y	No. Observations:	751
Model:	SARIMAX(6, 0, 0)	Log Likelihood	374.645
Date:	Thu, 23 May 2024	AIC	-733.291
Time:	20:19:40	BIC	-696.319
Sample:	0	HQIC	-719.046
	- 751		
Covariance Type:	opg		

	coef	std err	z	P> z	[0.025	0.975]
intercept	3.4521	0.347	9.949	0.000	2.772	4.132
ar.L1	0.6967	0.027	25.367	0.000	0.643	0.751
ar.L2	0.0483	0.041	1.191	0.234	-0.031	0.128
ar.L3	-0.1579	0.045	-3.486	0.000	-0.247	-0.069
ar.L4	0.2253	0.033	6.848	0.000	0.161	0.290
ar.L5	-0.2806	0.026	-10.665	0.000	-0.332	-0.229
ar.L6	0.0594	0.032	1.844	0.065	-0.004	0.123
sigma2	0.0215	0.001	25.994	0.000	0.020	0.023

Ljung-Box (L1) (Q):	0.01	Jarque-Bera (JB):	535.00
Prob(Q):	0.94	Prob(JB):	0.00
Heteroskedasticity (H):	0.77	Skew:	0.26
Prob(H) (two-sided):	0.04	Kurtosis:	7.10

Fuente. Autoría propia

Figura 33

Métricas de rendimiento del modelo ARIMA.

```
forecast,conf_int = stepwise_model.predict(n_periods=30,return_conf_int=True)
forecast = pd.DataFrame(forecast,columns=['y'])
df_conf = pd.DataFrame(conf_int,columns= ['Upper_bound','Lower_bound'])
df_conf["new_index"] = range(752, 782)
df_conf = df_conf.set_index("new_index")
timeseries_evaluation_metrics_func(test, forecast)
```

Evaluation metric results:-
 MSE is : 0.026233455745819974
 MAE is : 0.13677340262619975
 RMSE is : 0.1619674527360975
 MAPE is : 1.59503476170413
 R2 is : 0.006082624191491193

Fuente. Autoría propia

Desarrollamos el modelo en SARIMA según las indicaciones del libro (B V Vishwas, 2020).

Figura 34

Implementación del algoritmo SARIMA.

SARIMA

```
def timeseries_evaluation_metrics_func(y_true, y_pred):
    def mean_absolute_percentage_error(y_true, y_pred):
        y_true, y_pred = np.array(y_true), np.array(y_pred)
        return np.mean(np.abs((y_true - y_pred) / y_true)) * 100
    print('Evaluation metric results:-')
    print(f'MSE is : {metrics.mean_squared_error(y_true, y_pred)}')
    print(f'MAE is : {metrics.mean_absolute_error(y_true, y_pred)}')
    print(f'RMSE is : {np.sqrt(metrics.mean_squared_error(y_true, y_pred))}')
    print(f'MAPE is : {mean_absolute_percentage_error(y_true, y_pred)}')
    print(f'R2 is : {metrics.r2_score(y_true, y_pred)}',end='\n\n')

def Augmented_Dickey_Fuller_Test_func(series , column_name):
    print (f'Results of Dickey-Fuller Test for column: {column_name}')
    dfctest = adfuller(series, autolag='AIC')
    dfoutput = pd.Series(dfctest[0:4], index=['Test Statistic', 'p-value', 'No Lags Used', 'Number of Observations Used'])
    for key,value in dfctest[4].items():
        dfoutput['Critical Value (%s)'%key] = value
    print (dfoutput)
    if dfctest[1] <= 0.05:
        print("Conclusion:====>")
        print("Reject the null hypothesis")
        print("Data is stationary")
    else:
        print("Conclusion:====>")
        print("Fail to reject the null hypothesis")
        print("Data is non-stationary")
```

Nota. Se modela el dataset mediante el algoritmo SARIMA para posteriormente comparar las métricas.

Figura 35

Estadísticas del modelo SARIMA.

```
Augmented_Dickey_Fuller_Test_func(df['Sales' ],'Sales')
```

```
Results of Dickey-Fuller Test for column: Sales
Test Statistic          -5.412786
p-value                  0.000003
No Lags Used            13.000000
Number of Observations Used  767.000000
Critical Value (1%)      -3.438905
Critical Value (5%)      -2.865316
Critical Value (10%)     -2.568781
dtype: float64
Conclusion:====>
Reject the null hypothesis
Data is stationary
```

Nota. Se muestran estadísticas después de correr el modelo SARIMA, como por ejemplo el valor p.

Figura 36

Métricas de rendimiento del modelo SARIMA.

```
X = df[['Sales']]
train, test = X[0:-30], X[-30:]

stepwise_model = auto_arima(train, start_p=1, start_q=1, max_p=7, max_q=7, seasonal=True, start_P=1, start_Q=1, max_P=7, max_D=7, max_Q=7, m=m,
                             d=None, D=None, trace=True, error_action='ignore', suppress_warnings=True, stepwise=True)

stepwise_model.summary()
forecast, conf_int = stepwise_model.predict(n_periods=30, return_conf_int=True)
df_conf = pd.DataFrame(conf_int, columns= ['Upper_bound', 'Lower_bound'])
df_conf["new_index"] = range(1229, 1259)
df_conf = df_conf.set_index("new_index")
forecast = pd.DataFrame(forecast, columns=['close_pred'])
forecast["new_index"] = range(1229, 1259)
forecast = forecast.set_index("new_index")
timeseries_evaluation_metrics_func(actualtest, forecast)
```

```
Best model: ARIMA(1,0,0)(0,0,3)[4] intercept
Total fit time: 21.218 seconds
Evaluation metric results:-
MSE is : 554594.4247322972
MSE is : 593.5825642870574
RMSE is : 744.7109672431965
MAPE is : 11.566876791133632
R2 is : 0.20996340237607158
```

Fuente. Autoría propia

Objetivo específico 4

4) Comparar la precisión y la exactitud de las predicciones generadas por cada modelo utilizando métricas como el error absoluto medio (MAE) y el error porcentual absoluto medio (MAPE).

Comparación de resultados

Tabla 2

Comparaciones métricas de los cuatro modelos.

Modelo	horizonte	mse	rmse	mae	mape	r ²
FaceBook Prophet Sin Feriados-Vacaciones	30 días	0,042068	0,205106	0,159263	0,018904	0,261623
FaceBook Prophet Con Feriados-Vacaciones	30 días	0,042112	0,205211	0,159111	0,018881	0,265667
ARIMA	30 días	0,026233	0,161967	0,136773	1,595034	0,006082
SARIMA	30 días	554594,4	744,7109	593,5825	11,56687	0,209963

Fuente. Autoría propia

Error Cuadrático Medio (MSE):

- Facebook Prophet sin feriados-vacaciones: 0,042068
- Facebook Prophet con feriados-vacaciones: 0,042112
- ARIMA: 0,026233
- SARIMA: 554594,4

El modelo ARIMA tiene el MSE más bajo, indicando mayor precisión en términos de error cuadrático medio. El MSE del modelo SARIMA es extremadamente alto, lo que sugiere un rendimiento deficiente.

Raíz del Error Cuadrático Medio (RMSE):

- Facebook Prophet sin feriados-vacaciones: 0,205106
- Facebook Prophet con feriados-vacaciones: 0,205211
- ARIMA: 0,161967
- SARIMA: 744,7109

El modelo ARIMA tiene el menor RMSE, indicando un mejor ajuste en comparación con los modelos de Facebook Prophet y SARIMA, que presenta un RMSE extremadamente alto.

Error Absoluto Medio (MAE):

- Facebook Prophet sin feriados-vacaciones: 0,159263
- Facebook Prophet con feriados-vacaciones: 0,159111
- ARIMA: 0,136773
- SARIMA: 593,5825

El MAE del modelo ARIMA es más bajo, indicando menores errores absolutos en sus predicciones. El MAE de SARIMA es extremadamente alto.

Error Absoluto Porcentual Medio (MAPE):

- Facebook Prophet sin feriados-vacaciones: 0,018904
- Facebook Prophet con feriados-vacaciones: 0,018881
- ARIMA: 1,595034
- SARIMA: 11,56687

Los modelos de Facebook Prophet tienen un MAPE significativamente más bajo, lo que significa que tienen menores errores relativos porcentuales que los modelos ARIMA y SARIMA, siendo SARIMA el peor en esta métrica.

Coefficiente de Determinación r^2 :

- Facebook Prophet sin feriados-vacaciones: 0,261623
- Facebook Prophet con feriados-vacaciones: 0,265667
- ARIMA: 0,006082
- SARIMA: 0,209963

Los modelos de Facebook Prophet tienen un r^2 significativamente mayor que ARIMA y SARIMA, indicando que los modelos de Prophet explican una mayor proporción de la varianza en los datos.

Discusión sobre las Diferencias

Estructura de los Datos: Prophet está diseñado para manejar componentes de tendencia y estacionalidad de manera robusta (Hyndman, 2021), lo que lo hace adecuado para datos con patrones claros. ARIMA, aunque es un modelo fuerte para series temporales, puede tener dificultades si los datos tienen estructuras complejas no capturadas bien por sus componentes AR y MA. SARIMA, a pesar de su capacidad para manejar estacionalidades, podría estar mal especificado, lo que lleva a sus pobres resultados.

Componentes Estacionales y de Tendencia: Prophet maneja estos componentes de manera explícita y flexible, mientras que ARIMA y SARIMA requieren una correcta especificación de los parámetros. Esto hace que Prophet sea más robusto en situaciones donde los patrones no son fáciles de modelar con precisión.

Incorporación de Feriados y Vacaciones: Aunque intuitivamente debería mejorar el modelo, la incorporación de feriados y vacaciones en Prophet no ha mostrado una mejora significativa en este caso. Esto podría sugerir que los feriados y vacaciones no introducen variaciones suficientemente significativas en los datos para afectar las predicciones.

Especificación del Modelo: La mala especificación de SARIMA (elección incorrecta de los parámetros de estacionalidad, diferenciación, etc.) puede llevar a un desempeño extremadamente pobre. Es crucial realizar un análisis exhaustivo de los datos y pruebas para elegir los parámetros adecuados.

Ruido en los Datos: Si los datos contienen mucho ruido, ARIMA y SARIMA pueden tener dificultades para separar las señales relevantes, lo que afecta su desempeño (Hyndman, 2021). Prophet, por su diseño, es más robusto frente a tales situaciones.

Conclusiones

La tabla comparativa proporcionada evalúa el rendimiento de cuatro (4) modelos de pronóstico de series temporales: Facebook Prophet sin Feriados, ARIMA, Facebook Prophet con Feriados y SARIMA. Los modelos se evaluaron en función de diferentes métricas, incluyendo MSE, RMSE, MAE, MAPE y r^2 .

Los modelos de Facebook Prophet tienen un r^2 mucho mayor que el modelo ARIMA y SARIMA. Un mayor valor de r^2 indica un mejor ajuste del modelo a los datos reales. Esto significa que los modelos de Facebook Prophet explican una mayor proporción de la varianza en los datos.

En conclusión, si la precisión absoluta de las predicciones es el principal objetivo (MSE, RMSE, MAE), el modelo ARIMA es preferible. Sin embargo, si la precisión relativa (MAPE) y el ajuste general del modelo r^2 son más importantes, los modelos de Facebook Prophet son superiores. La inclusión de feriados y vacaciones en el modelo Facebook Prophet tiene un impacto muy leve en el rendimiento.

Bibliografía

- Aleksei Kychkin, G. C. (febrero de 2024). *AI-Powered Predictions for Electricity Load in Prosumer Communities*. <https://arxiv.org/abs/2402.13752>.
- Aristizábal Fúquene, J. A. (2017). *Diseño y aportes de un modelo para minería de*.
<https://repository.usta.edu.co/bitstream/handle/11634/3945/Aristizabaljorge2017.pdf?sequence=1>.
- B V Vishwas, A. P. (2020). *Hands-on Time Series Analysis with Python*. En A. P. B V Vishwas. Apress.
- Calabria, J. B. (2014). *Minería de datos, aplicación didáctica*. Barranquilla: Uniautónoma.
- Dieckmann, J. (24 de enero de 2024). *Getting Started Predicting Time Series Data with Facebook Prophet*. <https://towardsdatascience.com/getting-started-predicting-time-series-data-with-facebook-prophet-c74ad3040525>
- Emir Zunic, K. K. (7 de mayo de 2020). *Application of Facebook's Prophet Algorithm for Successful Sales Forecasting Based on Real-world Data*.
<https://arxiv.org/abs/2005.07575>
- Han Jiawei, K. M. (2014). *Data Mining Concepts and Techniques Third Edition*. Elsevier Science.
- Hernández Guerrero, J. I. (2018). *Sistema para la Enseñanza de Matemáticas basado en Evaluación Formativa*. Mexico: Cenid.
- Hyndman, R. &. (2021). *Forecasting: principles and practice*.

Li, S. a. (febrero de 2023). *Time series forecasting methods and their applications to particle accelerators*.

https://www.researchgate.net/publication/368552705_Time_series_forecasting_methods_and_their_applications_to_particle_accelerators

Luyao Zhang, F. Z. (mayo de 2023). *Understand Waiting Time in Transaction Fee Mechanism: An Interdisciplinary Perspective*. <https://arxiv.org/abs/2305.02552>

Md Rashidul Hasan, M. A. (14 de marzo de 2022). *A Comparative Study on Forecasting of Retail Sales*. <https://arxiv.org/abs/2203.06848>

Mehandzhiyski, V. (27 de abril de 2023). *How To Pre-Process Time Series Data*.

<https://365datascience.com/tutorials/time-series-analysis-tutorials/pre-process-time-series-data/>

Noemi, Á. F. (2018). *Técnicas de Machine Learning y Desarrollo de Modelos Predictivos Aplicados a la Antropología Forense*.

<https://openaccess.uoc.edu/bitstream/10609/82140/6/nalvarezfernTFM0618memoria.pdf>

Oscar Mauricio Bedoya Herrera, M. L. (2019). *Modelo Predictivo para la Identificación de Factores Socioculturales Asociados al Tiempo de Búsqueda del Primer Empleo en Egresados Universitarios*.

<https://revistavirtual.ucn.edu.co/index.php/RevistaUCN/article/view/1185/1563>

Romanchuk, J. (24 de octubre de 2023). *Machine Learning and Marketing: Tools, Examples, and Tips Most Teams Can Use*. <https://blog.hubspot.com/marketing/machine-learning-and-marketing>

- Sánchez, M. C.-M. (2013). *Modelos Predictivos para la Produccion de Productos Forestales no Maderables: Resina de Pino*.
https://www.researchgate.net/publication/279180188_Modelos_predictivos_para_la_produccion_de_productos_forestales_no_maderables_Resina_de_Pino
- Shu Wei Chou-Chen, L. A. (enero de 2024). *Forecasting hospital discharges for respiratory conditions in Costa Rica using climate and pollution data*.
<https://arxiv.org/abs/2401.03101>
- Taylor SJ, L. B. (2017). *Forecasting at scale*. <https://peerj.com/preprints/3190v2/>
- Witten, I. H. (2009). *Data Mining: Practical Machine Learning Tools and Techniques (2nd Edition)*. . Cambridge University Press.
- Zunic, E. a. (abril de 2020). *Application of Facebook's Prophet Algorithm for Successful Sales Forecasting Based on Real-world Data*.
https://www.researchgate.net/publication/341071241_Application_of_Facebook's_Prophet_Algorithm_for_Successful_Sales_Forecasting_Based_on_Real-world_Data

Apéndices

Apéndice A [Base de datos](#)

Apéndice B [Cuaderno *.ipynb donde se desarrolló el trabajo](#)

Apéndice C [Fuente de origen de los datos](#)