Universitat Oberta de Catalunya (UOC)

Máster Universitario en Ciencia de Datos (*Data Science*)

TRABAJO FINAL DE MÁSTER

Área: Data Science en el ámbito industrial

Previsión de demanda mediante técnicas de machine learning

Autor: Sebastián Calvo Martucci

Tutor: Lorena Polo Navarro

Profesor: Ismael Benito-Altamirano

A Coruña, 20 de junio de 2024

Créditos/Copyright

FICHA DEL TRABAJO FINAL

Título del trabajo:	Previsión de demanda mediante técnicas de machine learning
Nombre del autor:	Sebastián Calvo Martucci
Nombre del colaborador/a docente:	Lorena Polo Navarro
Nombre del PRA:	Ismael Benito Altamirano
Fecha de entrega (mm/aaaa):	06/2024
Titulación o programa:	Máster universitario en Ciencia de Datos
Área del Trabajo Final:	Data Science en el ámbito industrial
Idioma del trabajo:	Español
Palabras clave	Gestión de demanda, Modelo predictivo, Machine learning

Dedicatoria

A mis padres, por habérmelo dado todo.

Agradecimientos

A mi directora, Lorena Polo, por haberme guiado durante todo este proyecto. A mis compañeros de prácticas y de empresa, por su ayuda durante estos años. A mis amigos, que me dieron soporte en los momentos más complicados. A mi familia, pero sobre todo a mis padres, Daniel y Bettina, por su apoyo incondicional, no solamente durante el máster, si no durante toda la vida. A Flik, mi eterna compañía. A Teodora, un pilar de mi vida y referente por luchar por lo que más quieres. Gracias a todos.

Abstract

Stock management is crucial for a company's operational performance, facing challenges ranging from supplier issues to unexpected changes in demand and market promotions. This project tackles this complexity by creating a predictive model to estimate demand more accurately and facilitate stock management.

To achieve this goal, the project begins with the justification and contextualization, establishing objectives and the working methodology. Then, existing literature on proposed solutions is reviewed to provide a foundation. Additionally, a technique for estimating inventory calculations is investigated.

Following this, data from a company that sells automotive products is analyzed, involving data exploration and cleaning to ensure quality. Correlations between variables are examined, features are created, and clustering techniques are applied.

Once the exploration and preprocessing stages are completed, model selection proceeds, detailing the accuracy indicators and hyperparameters used.

Subsequently, the accuracy results from the different selected models are evaluated and ranked.

Finally, the economic impact of the improved algorithm is studied, and the project's conclusions are presented, including lessons learned, achievement of objectives, and potential future work directions.

Key words: Stock Management, Predictive Model, Machine Learning, Predictive Algorithms.

Resumen

La gestión del inventario es crucial para el rendimiento operativo de una empresa, enfrentando desafíos que van desde problemas con proveedores hasta cambios inesperados en la demanda y promociones del mercado. Este proyecto aborda esta complejidad mediante la creación de un modelo predictivo para estimar con mayor precisión la demanda y facilitar la gestión del stock.

Para lograr esta meta, se comienza con la justificación y contextualización del proyecto, estableciendo los objetivos y la metodología de trabajo. Posteriormente se revisa la bibliografía existente en cuanto a soluciones propuestas para tener un punto de partida. Además, se investiga sobre una técnica de estimación del cálculo de inventario.

Luego, se analizan los datos de una empresa de venta de productos de automoción, llevando a cabo la exploración y limpieza de los datos para garantizar su calidad. Se buscan correlaciones entre variables, se crean *features* y se aplican técnicas de *clustering*.

Una vez completada la etapa de exploración y preprocesamiento, se procede con la selección de modelos, explicando los indicadores de precisión e hiperparámetros utilizados.

Posteriormente se evalúan y clasifican los resultados de precisión obtenidos de los diferentes modelos seleccionados.

Finalmente, se estudia el impacto económico de la mejora del algoritmo y se presentan las conclusiones del proyecto, junto con las lecciones aprendidas, el logro de los objetivos y las posibles líneas de trabajo futuro.

Palabras clave: Gestión de demanda, Modelo Predictivo, Machine learning, Algoritmos predictivos.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

1.1. Contextualización y justificación

En la actualidad, el entorno empresarial se encuentra en una creciente demanda globalizada debido a un consumismo en constante aumento. Esta situación ejerce una presión considerable en la cadena de suministro, conviertiendo la logística en un reto cada vez más complejo. Esto provoca que la automatización de los procesos sea crucial ya que el impacto que puede suponer sobre el negocio es cada vez más crítico. Debido a esto, es necesario obtener la máxima eficiencia posible, innovando en todas las áreas y departamentos.

Por ello surge la necesidad de crear modelos predictivos que mejoren la precisión de la demanda y evitar inconvenientes como pueden ser el exceso de stock o la rotura de stock, situaciones que impactan negativamente en la rentabilidad del negocio.

En este proyecto se desarrolla un modelo predictivo que satisface la demanda y evita problemas logísticos.

1.2. Motivación

Actualmente me encuentro trabajando en proyectos de analítica de datos en el sector retail de moda, en concreto en departamentos de logística, por lo que la temática del inventariado es algo que encuentro interesante. Además, el hecho de poder estimar la demanda de los productos me parece fascinante debido a que nos encontramos en un mundo cuya tecnología evoluciona rápidamente y siempre debemos estar a la vanguardia de los avances. Poder saber con gran precisión la demanda necesaria es algo que hace 20 años no nos podíamos imaginar, por lo que es un gran desafío poder lograr estimaciones más precisas.

1.3. Objetivos

A continuación se describen los objetivos concretos de este trabajo de final de máster:

1.3.1. Objetivo principal

Este trabajo tiene como propósito principal **el desarrollo de un algoritmo** que ayude a estimar con precisión la demanda necesaria y evitar problemas como el sobrestock o rotura de stock.

Para conseguir este objetivo, son también necesarias las siguientes metas a llevar a cabo.

1.3.2. Objetivos secundarios

En esta sección se detallan los objetivos secundarios que complementan el propósito principal del trabajo:

1. Planificación de un proyecto de modelado predictivo.
2. Revisión bibliográfica del estado del arte sobre la problemática a tratar.
3. Reunir y organizar los datos que disponemos.
4. Llevar a cabo un análisis exploratorio EDA para investigar las características de los datos.
5. Encontrar los factores determinantes para predecir la demanda.
6. Realizar pruebas con diferentes algoritmos de *machine learning* hasta encontrar el que mejor resultados ofrece.
7. Comparar los algoritmos y clasificar sus resultados.
8. Análisis del impacto económico.
9. Documentar todo el procedimiento.
10. Realizar una presentación donde se exponga todo el proyecto.
11. Defender el proyecto.

1.4. Impacto en la sostenibilidad, responsabilidad social y diversidad. Objetivos de Desarrollo Sostenible

En relación a la **sostenibilidad**, este trabajo busca un enfoque en el que la predicción de la demanda busque reducir el desperdicio al garantizar que las empresas produzcan la cantidad adecuada de productos y evitar la sobreproducción. De esta manera, se minimiza el impacto ambiental.

Sobre la **responsabilidad social**, este trabajo quiere conseguir mediante predicciones precisas evitar situaciones de escasez que puedan afectar a los consumidores. Además, de esta manera se propicia la competición entre empresas, provocando precios más competitivos

En cuanto a la **diversidad**, este trabajo no está estrechamente relacionado con variables que puedan afectar a grupos demográficos y culturales, por lo que no habrá ningún tipo de sesgo.

Este proyecto se alinea con los objetivos de contribuir a la Agenda 2030 desde las principales funciones universitarias [1] al desarrollar un modelo predictivo que cumpla con la sostenibilidad, responsabilidad social y diversidad. Además, se compromete a seguir adelante con los Objetivos de Desarrollo Sostenible, concretamente con el **objetivo 9 (Industria, Innovación e Infraestructura)** y **12 (Producción y Consumo Responsables)**, ya que el resultado de este proyecto puede mejorar los procesos industriales, la gestión de inventario y la reducción de desperdicio de productos.

1.5. Recursos y metodología

Para realizar este proyecto, se utilizaran los siguientes recursos *software* y *hardware*:

- **Visual Studio Code y Google Colab**: IDE de desarrollo para realizar los EDA y la programación de los modelos predictivos. Python será el lenguaje de desarrollo y se utilizarán los entornos de Jupyter Notebooks.
- **Overleaf**: herramienta de redacción en lenguaje *LaTeX* online.
- **Gantt Project**: herramienta de planificación con diagramas de Gantt.
- **Google Chrome**: navegador web.
- **Github**: plataforma de desarrollo basado en el sistema de control de versiones git.
- **Microsoft Excel y Notepad++**: herramientas para la gestión inicial y análisis visual de los datos.

- **Microsoft OneNote**: herramienta para tomar notas e ideas.
- **Microsoft PowerPoint**: herramienta para desarrollar la presentación.
- **Lenovo Y530**: portátil para realizar el proyecto, así como toda su documentación. Sus especificaciones son las siguientes: Intel Core i7-8750H, 16GB de RAM, 1TB + 256GB de SSD y tarjeta gráfica GTX 1050.
- **Apple Ipad**: herramienta para analizar el estado del arte del proyecto, así como para tomar notas.

En cuanto a la metodología empleada, se desarrolla con CRISP-DM [12], que divide el proceso de minería de datos en seis fases principales (ver figura 1.1). La primera de ellas es la de **comprensión del negocio**. En esta fase, se tendrá que evaluar la situación actual de la gestión del inventario y cómo los diversos inconvenientes, como pueden ser las rotura de stock, pueden afectar en gran medida a la organización. Durante esta etapa también se revisa documentación acerca de los diferentes algoritmos de predicción que se utilizan para llevar a cabo las estimaciones sobre la demanda. Los resultados de estos análisis se documentan en este proyecto.

A continuación le sigue la **comprensión de los datos**. En esta fase, se organizan los conjuntos de datos que disponemos, se clasifican y se realizan tareas de exploración de los datos. Además, en esta fase se identifican posibles problemas en los datos para evitar incongruencias a la hora de preparar un modelo de datos. Las herramientas que se utilizan son principalmente un IDE como Visual Studio Code y el lenguaje de programación Python. El resultado de este análisis es un conjunto de datos preparado para la siguiente fase, en la que se seleccionan aquellos datos que aporten información.

La siguiente fase se corresponde con la **preparación de los datos**. En esta fase se separan los datos relevantes para crear un conjunto de datos que sea ideal para las siguientes fases. Además, se realizan tareas de limpieza como pueden ser eliminación de valores atípicos o errores de entrada. Además se considera la posibilidad de integrar otras fuentes de datos si fuese necesario. Una vez realizados estos pasos, se valora utilizar una técnica de reducción de la dimensionalidad para mantener solo aquellas componentes principales. El resultado de este proceso es un conjunto de datos listo para hacer pruebas de predicción con diferentes algoritmos y crear un modelo predictivo.

A continuación se realiza la fase de **modelado**, en la que se seleccionan diferentes algoritmos de predicción como pueden ser series temporales, regresión, árboles de decisión o redes neuronales. Además, se modifican los hiperparámetros en los diferentes entrenamientos buscando la mejor combinación posible. Durante esta fase se utilizan múltiples librerías de Python con los diferentes algoritmos. El resultado de esta fase es el conjunto de modelos entrenado.

A continuación viene la fase de **evaluación** en la que se analizan los resultados de los entrenamientos y se verifica el grado de acercamiento de los modelos a los objetivos de inventario. Se utilizan métricas de evaluación como el RMSE (Error Cuadrático Medio de la Raíz). Una vez finalizada esta fase, se consideran los resultados y si no son satisfactorios se puede iterar sobre las fases anteriores.

La última fase se corresponde con el **despliegue**, en el que se plantea un plan de despliegue del modelo y se lleva a cabo un seguimiento sobre los resultados que ofrece con el mundo real. En esta fase se proponen diversos puntos a seguir para evolucionar el modelo y que se adapte a la realidad.

Cabe destacar que toda esta metodología es **iterativa**, por lo que es probable que se realicen múltiples iteraciones sobre todo el proceso hasta encontrar resultados satisfactorios.

Figura 1.1: Gráfico de la metodología CRISP-DM. Fuente [13]

1.6. Planificación

La realización de este proyecto cuenta con cinco fases que tienen sus hitos correspondientes en forma de entregas. A continuación se explica cada una de ellas:

1. **PEC1 — Definición y planificación del trabajo final**: durante esta primera etapa se define la temática del trabajo, así como el contexto y relevancia. También se especifican los objetivos a cumplir y la planificación del proyecto.

2. **PEC2 — Estado del arte o análisis de mercado del proyecto**: en esta segunda fase se investiga sobre la temática escogida. Además, se revisa la bibliografía existente y otras posibles soluciones que se hubiesen desarrollado.

3. **PEC3 — Diseño e implementación del trabajo**: durante esta tercera fase se lleva a cabo el desarrollo del objetivo principal del proyecto.

4. **PEC4 — Redacción de la memoria**: en esta fase se refina y finaliza lo que hubiese quedado pendiente en la memoria.

5. **PEC5 — Presentación y defensa del proyecto**: en esta fase se realiza una breve presentación que se entrega junto a la memoria al tribunal. Además, se defiende de manera pública.

A continuación se muestra en la figura 1.2 la planificación con un diagrama de Gantt:

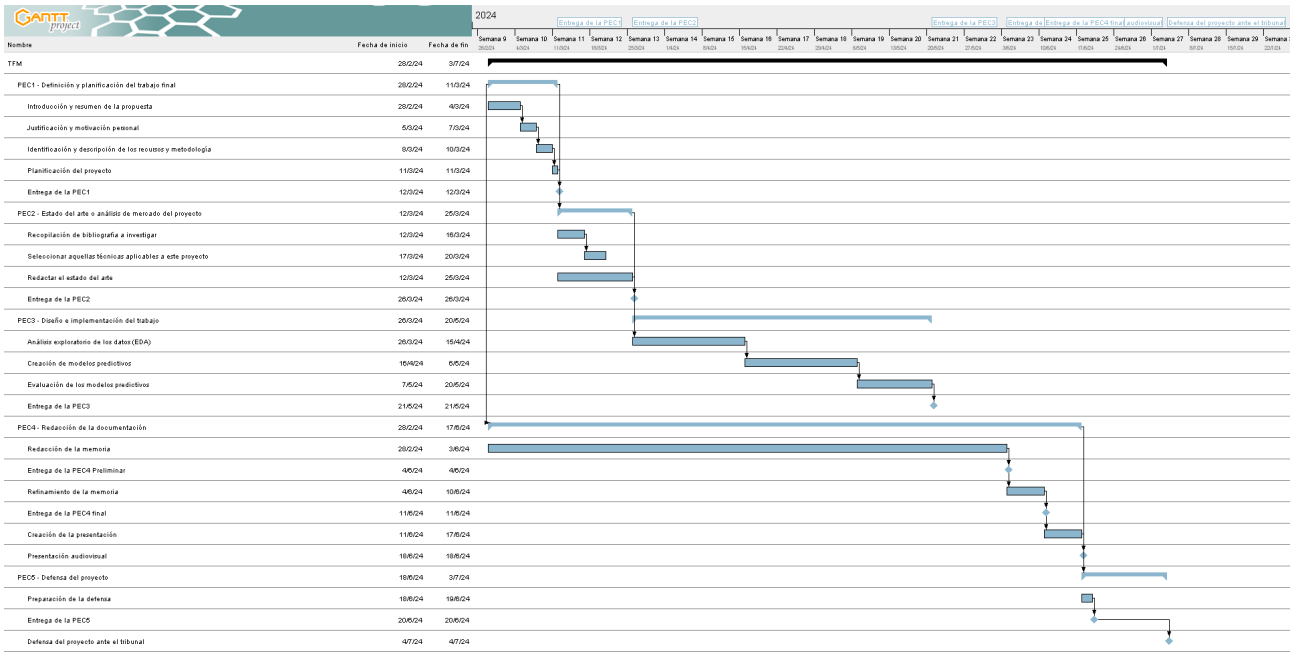

Figura 1.2: Diagrama de Gantt.

Capítulo 2

Estado del arte

Durante el siguiente capítulo se explica en detalle los factores causantes de los problemas que existen en cuanto a la gestión de inventario y demanda y que suponen una incógnita en múltiples empresas.

En este proyecto se busca la eficiencia en los almacenes mejorando la precisión de la demanda, facilitando la gestión del inventario y evitando los siguientes problemas que se van a detallar a continuación.

2.1. Problemática y estado de la gestión del inventario y demanda

Cuando se habla en términos de inventario, siempre se busca encontrar el equilibrio entre tener la cantidad necesaria para no tener ningún problema en la demanda y el no malgastar recursos por falta de demanda.

Por ello, existe un conflicto entre tener un **inventario alto** o un **inventario bajo**. Por un lado, cierta influencia en mantener un inventario bajo debido a que representa una inversión monetaria temporal en bienes. Además, el coste de manejar el mismo involucra otro tipo de costes como pueden ser el de interés, el de oportunidad, almacenamiento, impuestos, seguros y riesgos de obsolescencia. Tratar de mantener un inventario bajo puede ser una prioridad en una empresa cuando su objetivo es optimizar los recursos de los que dispone y mejorar su eficiencia operativa.

Por otro lado puede existir cierta presión en mantener un inventario alto, ya que se puede priorizar la necesidad de garantizar un servicio al cliente satisfactorio asegurando la disponibilidad de los productos. Manteniendo un inventario alto se pueden reducir ciertos costes asociados a los pedidos, como pueden ser mano de obra y transporte.

Se busca de tener un equilibrio de forma que la gestión del inventario requiera una inversión mínima manteniendo un buen servicio al cliente. En esta gestión, se producen varios tipos de costes a tener en cuenta [24]:

- **Coste de posesión o mantenimiento:** alguno de los costes asociados a la gestión del inventario son los relacionados con el almacenamiento, desperdicios, daños, seguros, obsolescencia, impuestos o despreciación. Este tipo de costes tienden a reducir la inversión y los riesgos financieros cuando se sigue la estrategia de mantener niveles bajos de inventario y realizar reposiciones frecuentes.
- **Coste de emisión de pedidos:** este tipo de costes incluyen los gastos administrativos y de oficina que se asocian a la preparación de órdenes de compra.
- **Coste de adquisición:** estos costes incluyen los gastos asociados con la obtención del material, configuración del equipo y papeleo. Reducir este tipo de coste es un objetivo importante para permitir la fabricación de lotes más pequeños y de esa manera reducir el inventario.
- **Coste de ruptura:** este tipo de coste sucede cuando un artículo se agota, incurriendo en retrasos o cancelaciones. Suele ser un coste difícil de controlar, ya que es complejo mantener un equilibrio entre tener un inventario adecuado y los costes asociados, por lo que estimar las pérdidas de ganancias o la insatisfacción del cliente se vuelve complicado. Puede suceder tanto en suministros externos como pueden ser pérdida de ventas o daño de imagen, como en suministros internos como paradas en la producción, horas extra o subcontratas.

Además de los costes directos asociados a la gestión del inventario, existen otro tipo de riesgos relacionados con el **tiempo** que pueden afectar a la efectividad de la organización:

- **Tiempo de confección del pedido (documentación)**: es el tiempo necesario para preparar la documentación de los pedidos.
- **Tiempo de desplazamiento o transporte**: como su nombre indica, se refiere al tiempo necesario para el transporte del producto desde origen hasta destino.
- **Tiempo de cola**: es el tiempo en el que el producto permanece en una lista de espera.
- **Tiempo de preparación**: se refiere al tiempo requerido para la configuración y preparación de los recursos utilizados en la cadena de producción.
- **Tiempo de ejecución**: es el tiempo de procesamiento de un producto.

- **Tiempo de espera**: se refiere al tiempo en el que el producto permanece inactivo hasta que llegue a la siguiente fase.
- **Tiempo de inspección**: es el tiempo empleado en el control de calidad del producto.

Otro de los factores importantes a la hora de gestionar la demanda sería el de **demanda independiente y dependiente**. Cuando las necesidades de los productos no están relacionadas, se considera que son independientes ya que no surge la necesidad de mantener el stock de ambos productos. En el caso de ser dependientes, se debe tener en cuenta este aspecto a la hora de gestionar el stock para no sufrir rotura de stock. En la siguiente figura 2.1 se puede observar un marco de trabajo que muestra cómo las características de la demanda, el coste de las transacciones y el riesgo de obsolescencia pueden afectar al sistema de control de inventario.

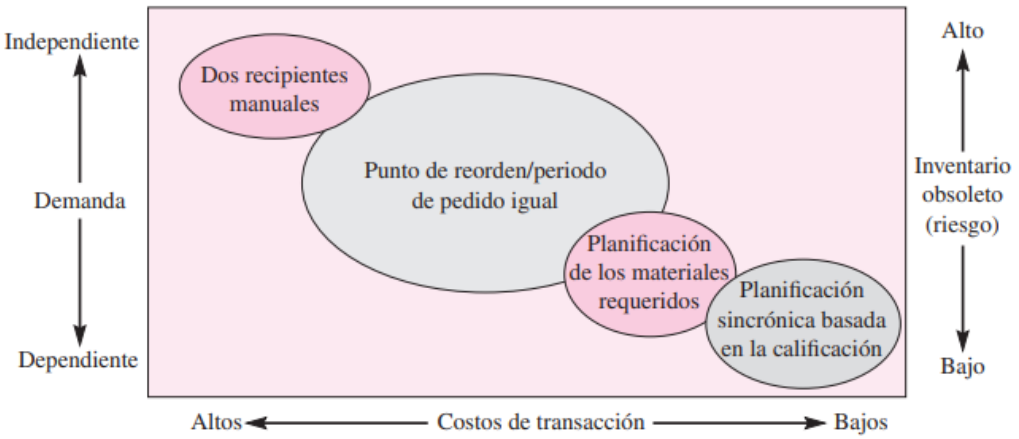

Figura 2.1: Marco que describe la lógica del control de inventarios. Fuente [24]

Todos estos factores que se han explicado hasta el momento deben tenerse en cuenta en los **sistemas de inventario**.

Existen dos tipos de sistemas: los de **periodo único**, dirigidos a empresas que hacen pedidos de temporada o solo una vez y se suele aplicar en situaciones de artículos de temporada o perecederos y los de **varios periodos**, dirigidos a empresas que quieren garantizar la disponibilidad del producto durante todo el año. Ambos modelos se rigen según el grado de conocimiento y de la tasa de la demanda, por lo que pueden ser **modelos deterministas**, donde la demanda es conocida (pueden ser una tasa constante o variable) o **modelos aleatorios**, donde la demanda y/o el tiempo son variables aleatorias.

Entrando en detalle en los sistemas de inventario de varios pedidos, se pueden clasificar en dos tipos (se pueden observar sus diferencias en la figura 2.2):

- **Modelos de cantidad de pedido fija (*sistema Q*)**: se realizan pedidos de un tamaño fijo dependiendo del nivel de inventario y de la demanda prevista, pudiendo hacerse en

cualquier momento. Suele utilizarse para productos costosos, ya que compensa los gastos derivados por el mayor control que requiere.

- **Modelos de periodo fijo (*sistema P*)**: se establece un intervalo de tiempo constante entre cada pedido, los cuales se efectúan al final de cada uno y su tamaño varía según el nivel de inventario y de la demanda esperada en ese momento. Suele utilizarse para productos de poco valor debido a su mayor necesidad de inventario.

Característica	*Modelo Q* Modelo de cantidad de pedido fija	*Modelo P* Modelo de periodo fijo
Cantidad del pedido	*Q*, constante (siempre se pide la misma cantidad)	*q*, variable (varía cada vez que se hace un pedido)
Dónde hacerlo	*R*, cuando la posición del inventario baja al nivel de volver a pedir	*T*, cuando llega el periodo de revisión
Registros	Cada vez que se realiza un retiro o una adición	Solo se cuenta en el periodo de revisión
Tamaño del inventario	Menos que el modelo de periodo fijo	Más grande que el modelo de cantidad de pedido fija
Tiempo para mantenerlo	Más alto debido a los registros perpetuos	
Tipo de pieza	Piezas de precio más alto, críticos o importantes	

Figura 2.2: Diferencias entre los sistemas. Fuente [24]

Como se puede observar, existen numerosas incógnitas a la hora de gestionar un inventario. Por ello, es necesario contar con estimaciones de demanda precisas para llegar a ser más eficientes en la gestión.

2.2. Revisión bibliográfica acerca de métodos de estimación de la demanda

En la siguiente sección se analiza la labor realizada por otros investigadores en el área para conseguir predicciones de demanda mediante **técnicas de machine learning** y alinear mejor el stock a la demanda.

Los investigadores Huber y Stuckenschmidt realizaron un artículo [21] en el que tratan la problemática de la estimación de la demanda y prueban diferentes algoritmos. En él, plantean el pronóstico de la demanda para los minoristas, haciendo énfasis en la previsión en **días especiales**, ya que están sujetos a patrones de demanda muy diferentes que en días regulares. El caso concreto es el de una panadería y se aborda el problema de pronosticar la demanda diaria para diferentes categorías de productos a nivel de tienda. Utilizaron técnicas de machine learning supervisado, así como redes neuronales y árboles de decisión potenciados por gradiente.

Las pruebas realizadas mostraron que las **técnicas de machine learning** son una **alternativa viable** para conseguir estimar la demanda, ya que consiguen mejores resultados que

técnicas clásicas como series temporales o regresión, sobre todo cuando se quiere predecir la demanda en días especiales, ya que los resultados tienen un entre un 10 y un 20 por ciento menos de error. Esto se debe a que se incorporan características de los días especiales, lo que hace que los ajustes de preprocesado y postprocesado de los otros algoritmos queden obsoletos. Además, no requiere construir múltiples modelos de datos, se puede entrenar uno global que sea lo suficientemente preciso.

Por otro lado, llegaron a la conclusión de que el **reentrenamiento** favorece la precisión entre un uno y un tres por ciento y que cuando se predice a un futuro distante, el error es estable durante las temporadas, por lo que los métodos evaluados son constantes en diferentes períodos.

También observaron que las ***RNN*** (Red Neuronal Recurrente), concretamente las del tipo ***LSTM***, superan a las ***ANN*** (Red Neuronal Artificial), en concreto el ***MLP*** (Perceptrón Multicapa) y los **árboles potenciados por gradiente *GBRT***. Además comprobaron que es mejor tratar el problema como uno de **clasificación** que como uno de regresión (en el caso de las ANN).

Los autores concluyen que, aunque los resultados son buenos, el uso de algoritmos sofisticados junto a la elección de hiperparámetros pueden mejorar aún más los resultados. Además, proponen una línea de investigación haciendo stacking de algoritmos.

Otro artículo que se ha explorado es el de Vinit Taparia et al. [18]. En él, tratan de evitar los algoritmos clásicos como *Naive Bayes* o la media móvil ya que no tienen buenos resultados cuando la tendencia es no lineal. El enfoque se centra en **utilizar diferentes algoritmos de regresión** (diferente al artículo de Huber [21]) e identifica el mejor por *SKU* con el menor error de precisión. Además, buscan crear un **modelo híbrido** combinando los **dos mejores algoritmos de regresión** para cada *SKU*.

Tras probar diferentes algoritmos como árboles de decisión, algoritmos de regresión lineal y polinómicos, **Random Forest** es el que mejor resultados ha dado con métricas como el *MAPE* (Error Porcentual Absoluto Medio), *MAD* (Desviación Absoluta Media) y *MSE* (Error Cuadrático Medio). En cuanto al híbrido, los mejores resultados fueron al utilizar la combinación **Random Forest con Regresión Polinómica**.

En conclusión, el **modelo híbrido** propuesto **mejora significativamente** la predicción de la demanda, por lo que los costes de inventario pueden reducirse, optimizando las decisiones de negocio. Aún así, este estudio tiene ciertas limitaciones importantes, ya que no tiene en cuenta ciertos parámetros fundamentales como días especiales, vacaciones, tiempo...

Otro de los artículos analizados fue el de Sai y Vedavath [23]. En él, estudian diferentes técnicas de machine learning para la predicción de ventas. Emplearon un dataset compuesto por datos de tiendas Rossmann, concretamente de más de 1000 tiendas, con un histórico de

casi tres años. Probaron algoritmos como *XGBoost*, Regresión Lineal o *ARIMA*, con datos que iban desde la información de tiendas hasta datos de clientes y localización. Tras las pruebas utilizando el RMSE en diferentes algoritmos, comprobaron que **XGBoost** era el que mejor resultados ofrecía.

También es interesante el artículo de Carbonneau [16]. En él, tratan de investigar diferentes técnicas de machine learning como redes neuronales y la aplicabilidad en cuanto a la predicción de la demanda, comparando estos métodos con otros más tradicionales como la tendencia, media móvil o regresión lineal. Los datos utilizados se corresponden con dos modelos diferentes: un conjunto de datos de una cadena de suministro **simulada** y otro de pedidos reales de fundiciones canadienses. Para el primero, desarrollaron una simulación en MATLAB Simulink, que modela retrasos de comunicación y entrega, así como la generación de demanda del cliente final con patrones de estacionalidad y ruido blanco. En la figura 2.3 se puede observar un diagrama con la simulación propuesta. Para el segundo modelo, utilizaron los datos de las ventas mensuales de todas las fundiciones clasificadas como industrias metaleras. Al estar en un puesto temprano en la cadena de suministro, están sujetas a una gran distorsión en la señal de demanda.

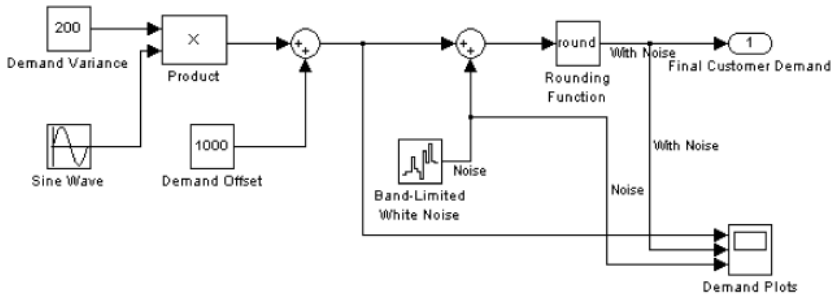

Figura 2.3: Diagrama con las diferentes señales utilizadas para la simulación. Fuente [16]

En cuanto a las pruebas, las técnicas de machine learning mostraron mejores resultados en general, pero no supusieron una gran mejora sobre las técnicas tradicionales (naive, tendencia, media móvil...) sobre el conjunto de datos simulado. En cambio, para el conjunto de datos real, las técnicas de machine learning *RNN* y *SVM* proporcionaron mejoras más significativas.

Con esta investigación concluyeron que la **regresión lineal múltiple** y el uso de técnicas de machine learning como **RNN y SVM** pueden mejorar las predicciones de demanda.

Otro artículo muy interesante es el de Adnan [14], ya que innova frente a los otros artículos utilizando **DeepAR**, un modelo de aprendizaje automático desarrollado por *AWS* (Amazon Web Services) [11] para pronosticar series temporales escalares (unidimensionales) utilizando redes neuronales recurrentes.

En la investigación utilizaron datos de diversas fuentes, como ventas, inventario y calendario

para la previsión. Estos datos son generados en tiempo real y se lleva a cabo una tarea de limpieza previa antes de ser procesados.

Una vez normalizados, se utilizan en el entrenamiento, escogiendo los hiperparámetros que tienen un gran impacto en los resultados. El algoritmo utilizado es el DeepAR, que se utiliza para predecir las ventas semanales de las tiendas basadas en series temporales.

Según las pruebas, los cálculos de pronóstico y rendimiento de los modelos DeepAR fueron un éxito. Los resultados fueron **altamente precisos**, con errores relativamente pequeños.

Otro artículo que vale la pena estudiar es el de Resul Tugay y Sule Gunduz Oguducu [25]. En él, centraron su investigación en la demanda de un sitio web de comercio electrónico que opera bajo un modelo de mercado competitivo con varios vendedores. Su propuesta fue la de aplicar diferentes algoritmos de regresión utilizando stacking para predecir la demanda y evaluando los diferentes enfoques.

El conjunto de datos contiene los datos de venta de cada producto del *e-commerce*, con su marca de tiempo. A este conjunto, se le aplican diferentes combinaciones de algoritmos hasta encontrar resultados satisfactorios.

Tras evaluar el modelo mediante RMSE y la prueba ANOVA, observaron que la regresión lineal destacó sobre múltiples algoritmos. Además, en el stacking binario, la combinación de GBT y RF tuvo un RMSE menor, y en el triple, **LR, RF y GBT**.

Con este trabajo, los autores concluyeron que hacer stacking de algoritmos mejoran las predicciones.

Otro artículo interesante a revisar es el de Takashi Tanizaki et al. [17] que busca implementar técnicas de predicción sobre la demanda en los restaurantes.

En él, destacan la necesidad de establecer un **modelo de predicción de demanda específico para cada tienda**, considerando factores como la ubicación, el clima y los eventos, por lo que utilizar técnicas de machine learning puede ser beneficioso.

En las pruebas utilizaron datos de cinco tiendas y la variable objetivo fue el número de visitantes según el mes, día de la semana, evento y según el clima. Evaluaron métodos como Regresión Lineal Bayesiana, Regresión de Árbol de Decisión Potenciado, Regresión de Bosque de Decisión y Método Stepwise. Los **resultados fueron similares**, siendo el Árbol de Decisión Potenciado el que menor precisión tenía.

En el artículo de Ariful Islam Arif et al. [15], proponen un método de predicción automática analizando los datos de una tienda considerando datos del mercado real, localización, temporada y eventos especiales.

En cuanto al método, los investigadores prueban diferentes algoritmos relevantes como *K-Nearest Neighbor, Support Vector Machine, Gaussian Naive Bayes, Random Forest y Decision Tree Classifier*. Concluyeron que, considerando factores como el comportamiento del cliente, el

clima estacional y las ocasiones especiales, el algoritmo con mayor precisión fue el de **Gaussian Naive Bayes**.

Otro artículo interesante e innovador es el de I-Fei Chen y Chi-Jie Lu [20]. En él, trataron de construir modelos de pronóstico de demanda para la industria de la moda, un gran desafío debido al *fast fashion* y los cambios de tendencia.

Este estudio utilizó un conjunto de datos de ventas mensuales de una marca famosa de *fast fashion*. Concretamente, los datos de las tiendas físicas clasificadas en dos tipos: mismas tiendas y nuevas tiendas. Utilizaron datos de demanda para construir modelos de pronóstico mediante *clustering* **KMeans** combinado con *ELM* (Extreme Learning Machine) o *SVR* (Support Vector Regression). Los resultados mostraron que **KMeans-ELM** y **KMeans-SVR** tienen una precisión de predicción más alta que los modelos de predicción ELM y SVR simples. Esto indica que los modelos de predicción basados en el *clustering* pueden mejorar las predicciones. Además, según el estudio, se recomienda tener en cuenta las **variables predictoras como días festivos, eventos importantes y condiciones económicas**.

El último de los artículos revisados es el de Shuojiang Xu y Hing Kai Chanb [22]. Es interesante ya que trata de predecir la demanda en la cadena de suministros para los servicios de atención médica, que supone un factor crítico. Por ello, los investigadores proponen un modelo de pronóstico de demanda de dispositivos médicos utilizando consultas en línea, basado en la correlación entre las consultas y la demanda real, lo que mejoraría la eficiencia de la cadena de suministro de atención médica. En la figura 2.4 se puede ver un diagrama con la propuesta.

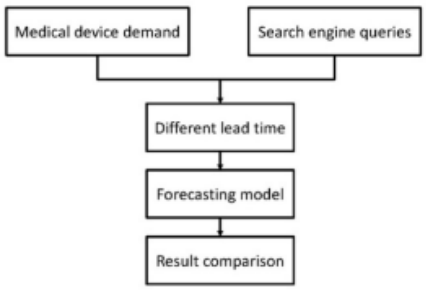

Figura 2.4: Diagrama con con la estructura del sistema de predicción. Fuente [22]

El conjunto de datos comprende los registros de demanda diaria de tres años de un producto vendido en 2000 hospitales. Calculan la demanda mensual del producto y recopilan las consultas en motores de búsqueda. Tras realizar diferentes pruebas con métodos como SVM, ARIMA o ANN, el algoritmo ganador fue el **SVM**.

A continuación se muestra una tabla con las conclusiones de las referencias bibliográficas analizadas (cuadro 2.1):

Autor (Año)	**Features utilizadas**	**Técnicas utilizadas**	**Métricas utilizadas**	**Técnica ganadora**
Carbonneau et al. (2008)	Datos simulados y reales de demanda	Naive, Media Móvil, Tendencia MLR, RNN, LV-SVM, NN	MAE, SD	**MLR, RNN, LS-SVM**
Resul Tugay y Sule Gunduz Oguducu (2009)	Datos de demanda en e-commerce de diferentes vendedores y calendario	DT, GBT, RF, LR	RMSE	**LR-RF-GBT**
Takashi Tanizaki et al. (2018)	Datos de 5 tiendas, calendario y tiempo	BLR, BDTR, FR, Stepwise	Precisión	**BLR, FR, Stepwise**
Ariful Islam Arif et al. (2019)	Datos del mercado real, localizacion, temporada y eventos especiales	KNN, RF, FNN, ANN, Holt-Winters, GNB	Precisión, MAPE	**Gaussian Naive Bayes**
Shuojiang Xu y Hing Kai Chanb (2019)	Datos de demanda de dispositivos médicos y de consultas online.	ARIMA, ANN, SVM	RMSE, MAPE	**SVM**
Adnan et al. (2020) Huber y Stuckenschmidt (2020)	Datos de demanda y calendario Datos relativos a la tienda, datos de transacciones, horario, festivos y datos de localización	DeepAR LIN-REG, ETS, LSTM-CL, MLP-CL, MLP-REG, LGBM	Precisión MAE, MASE	**DeepAR** **LSTM-CL**
Sai Ramya y K. Vedavath (2020)	Datos de ventas en tiendas, horario y rebajas	ARIMA, LR, RF, XGBoost	RMSE	**XGBoost**
I-Fei Chen y Chi-Jie Lu (2021)	Datos de demanda en tiendas y calendario	ELM, SVR, KMeans-ELM, KMeans-SVR	MAPE, RMSE	**KMeans-ELM**
Vinit Taparia et al. (2023)	Datos relativos de transacciones, calendarios, datos de competidores y datos de productos similares	DTR, SVR, LR-PR, LR-RF, LR-DTR, LR-SVR, PR-DTR, PR-SVR, PR-RF, RF-SVR, DTR-SVR, RF-DTR	MAPE, MAD, MSE	**PR-RF**
Sebastián Calvo Martucci (2024)	Datos históricos de ventas y stock por producto, además de promociones y calendarios.	Naive, RF, DT, LGBM, XGBoost, LSTM	EVS, MSE, RMSE	**RF-DT-LGBM-XGBoost-LSTM**

Cuadro 2.1: Tabla con información bibliográfica ordenada por año.

2.3. Aproximación de cálculo de inventario

Aunque este proyecto se centra en el cálculo de la demanda, es interesante analizar las ventajas que se pueden obtener gracias al modelo generado por algoritmos de machine learning. Por ello, se realiza el cálculo del stock en base a la demanda y se estima el impacto económico del modelo creado.

Yamazaki et al. [19] llegaron a la conclusión de que se puede estimar el stock de seguridad en base a la incertidumbre de la demanda y posteriormente calcular el impacto económico. De forma que la reducción del RMSE de los modelos de previsión conduce a una reducción del stock de seguridad necesario en el almacén, con la reducción de costes asociada que ello supone. Las ecuaciones utilizadas fueron las mostradas en las figuras 2.5 y 2.6, definidas a continuación:

$$\text{Stock de seguridad} = \text{Factor servicio} \times \text{RMSE} \times \sqrt{\text{ciclo de aprovisionamiento}}$$

Figura 2.5: Ecuación de stock de seguridad

donde:

- *Factor servicio* es la distribución normal estándar inversa para una probabilidad dada.
- *RMSE* es una estimación para la desviación estándar de la demanda.
- *Ciclo de aprovisionamiento* es la suma de días entre pedido y leadtime.

$$\text{Coste del stock de un día en stock} = \%\text{Coste unitario} \times \text{valor de las uds en stock}$$

Figura 2.6: Ecuación de coste de stock diario

donde:

- *Coste unitario* es el coste de almacenaje más el coste de oportunidad.
- *Valor de las uds en stock* es el producto de las unidades en stock por el precio del producto.

Capítulo 3

Diseño e implementación del trabajo

En este proyecto se siguen las etapas características de un proyecto habitual de ciencia de datos, utilizando la metodología CRISP-DM.

Como se explica en esta metodología, es un proceso iterativo, en el que las fases no son secuenciales, ya que se va perfeccionando el proceso a la vez que se va trabajando sobre el modelo de datos.

Además, durante el mismo se busca analizar con profundidad cada una de las variables, buscando características y correlaciones.

Asimismo, se trabaja en la problemática propuesta utilizando múltiples algoritmos de aprendizaje supervisado, clasificando los resultados de cada uno.

En el siguiente capítulo se detallan las diferentes fases en la implementación de este proyecto.

3.1. Origen de los datos

Durante esta sección se exponen los diferentes detalles relativos a los datos de los que se dispone para este proyecto.

3.1.1. Obtención de los datos

Los datos se corresponden con los de un caso real de un punto de venta de una empresa dedicada a la distribución de productos de automoción. Estos productos incluyen ambientadores, neumáticos, escobillas limpiaparabrisas, entre otros.

Estos datos fueron recogidos por mi tutora del proyecto, Lorena Polo Navarro. Además, cabe destacar que los datos han sido convenientemente anonimizados (no contiene ningún tipo de descripción y los identificadores fueron reemplazados) para la realización de este trabajo.

3.1.2. Descripción de los datos

Los datos reunidos están contenidos en los ficheros que se muestran en la siguiente tabla (cuadro 3.1):

Fichero	Descripción
01_Ventas	Ventas diarias por producto desde el 2021-04-03 hasta el 2023-04-03.
02_Calendarios	Calendario diario con un indicador que marca la apertura/cierre del punto de venta y un indicador que marca si era festivo o no. Desde el 2021-04-04 hasta el 2023-07-12.
03_Promociones	Periodos promocionales de los productos. Desde el 2011-01-04 hasta el 2023-05-17.
04_Stock	Stock diario por producto desde el 2021-04-04 hasta el 2023-04-04.
DatosCicloAprovisionamiento	Datos del ciclo de aprovisionamiento de los productos.
DatosPrecioMedio	Datos de precio medio de los productos.

Cuadro 3.1: Tabla con la descripción de los datos.

Estos ficheros contienen las siguientes variables:

- **Ventas**:
  - producto: identificador único del producto [1,1000].
  - idSecuencia: fecha de venta. Se cuenta con datos desde el 03-04-2021 hasta el 03-04-2023.
  - udsVenta: unidades vendidas. Se cuenta con datos negativos. Desde -327 uds hasta 510.
- **Calendarios**:
  - idSecuencia: dimensión de fecha. Se cuenta con datos desde el 04-04-2021 hasta el 12-07-2023.
  - bolOpen: indicador booleano. Si es 1, indica apertura del punto de venta, 0 cierre del punto de venta.
  - bolHoliday: indicador booleano. Si es 1, indica fecha festiva, 0 fecha no festiva.

- **Promociones**:
  - producto: identificador único del producto [1,1000].
  - idSecuenciaIni: fecha de inicio de promoción. Desde el 04-01-2011 hasta el 20-04-2023.
  - idSecuenciaFin: fecha de fin de promoción. Desde el 11-01-2011 hasta el 07-05-2023.
- **Stock**:
  - producto: identificador único del producto [1,1000].
  - idSecuencia: fecha registro de stock. Se cuenta con datos desde el 03-04-2021 hasta el 03-04-2023.
  - udsStock: unidades en stock. Se cuenta con datos negativos. Desde -149 uds hasta 2460.
- **DatosCicloAprovisionamiento**:
  - producto: identificador único del producto.
  - diasEntrePedidos: es el número de días entre pedidos.
  - diasLeadtime: es el número de días entre orden de compra y entrega al cliente.
- **DatosPrecioMedio**:
  - producto: identificador único del producto.
  - eurPrecioMedio: precio medio del producto.

3.1.3. Carga de los datos

Se cuenta con un fichero excel con los 4 conjuntos de datos principales (**Ventas**, **Calendarios**, **Promociones** y **Stock**) con los que se hace el análisis, preprocesado, modelado y entrenamiento de los datos y otros dos ficheros excel (**DatosCicloAprovisionamiento**, **DatosPrecioMedio**) que se utilizan en una fase posterior para evaluar el impacto económico del modelo final.

3.2. Exploración inicial de los datos

Durante esta fase se realiza un primer acercamiento de los datos, haciendo hincapié en el análisis estadístico y revisión de la calidad de los datos.

3.2.1. Resumen de los datos

El conjunto de datos final consta de cuatro tablas: **Ventas, Calendarios, Promociones y Stock**.

La tabla de **Ventas**, como su nombre indica, contiene un histórico de ventas con la fecha de venta (idSecuencia), el producto vendido (producto) y la cantidad de unidades vendidas (udsVenta).

La tabla **Calendarios** contiene los datos diarios desde el 04-04-2021 hasta el 12-07-2023 con dos indicadores, uno que indica la apertura y clausura del puesto de venta (bolOpen) y otro que indica la fecha festiva (bolHoliday).

La tabla **Promociones** contiene los datos de los periodos promocionales de los productos. Se cuenta con el producto y una fecha inicio (idSecuenciaIni) y una final (idSecuenciaFin) que indica el periodo promocional del producto.

La tabla **Stock**, como su nombre indica, contiene el histórico de stock con la fecha (idSecuecnai), el producto en stock (producto) y la cantidad de unidades en stock (udsStock).

Una vez finalizado en análisis individual de las tablas, se procede a unirlas mediante la clave de **producto** e **idSecuencia** (fecha), según la tabla.

3.2.2. Resumen estadístico de los datos

Se comienza el análisis estadístico inicial con la tabla de **Ventas** (figura 3.1):

	producto	idSecuencia	udsVenta
count	707608.000000	7.076080e+05	707608.000000
mean	499.500000	2.021821e+07	3.781253
std	289.284031	6.469619e+03	6.788273
min	1.000000	2.021040e+07	-327.000000
25%	247.750000	2.021100e+07	0.000000
50%	499.500000	2.022040e+07	0.000000
75%	750.250000	2.022100e+07	5.000000
max	1000.000000	2.023040e+07	510.000000

Figura 3.1: Estadísticos de Ventas.

Las estadísticas descriptivas de las variables producto, idSecuencia y udsVenta muestran que el conjunto de datos consta de 707608 registros. La variable producto tiene un rango de valores entre 1 y 1000. La variable udsVenta tiene una media de 3.78 unidades vendidas por registro y muestra una desviación estándar de 6.79, lo que sugiere una variabilidad significativa en las ventas. Además, se observa que udsVenta tiene un mínimo de -327, lo cual puede indicar

errores en los datos o devoluciones, y un máximo de 510 unidades vendidas en un solo registro. Cuenta con datos desde el 03-04-2021 hasta el 03-04-2023.

Análisis estadístico inicial de la tabla **Calendario** (figura 3.2):

	idSecuencia	bolOpen	bolHoliday
count	8.300000e+02	830.000000	830.000000
mean	2.021970e+07	0.855422	0.162651
std	7.276925e+03	0.351887	0.369269
min	2.021040e+07	0.000000	0.000000
25%	2.021103e+07	1.000000	0.000000
50%	2.022052e+07	1.000000	0.000000
75%	2.022122e+07	1.000000	0.000000
max	2.023071e+07	1.000000	1.000000

Figura 3.2: Estadísticos de Calendario.

Las estadísticas descriptivas para las variables idSecuencia, bolOpen y bolHoliday muestran un total de 830 registros. Cuenta con datos desde el 04-04-2021 hasta el 12-07-2023 y el resto de variables son booleanas.

Análisis estadístico inicial de la tabla de **Promociones** (figura 3.3):

	producto	idSecuenciaIni	idSecuenciaFin
count	26087.000000	2.608700e+04	2.608700e+04
mean	438.767010	2.018103e+07	2.018192e+07
std	293.472291	3.057438e+04	3.047187e+04
min	1.000000	2.011010e+07	2.011011e+07
25%	173.000000	2.016061e+07	2.016071e+07
50%	416.000000	2.018121e+07	2.019011e+07
75%	683.000000	2.021031e+07	2.021041e+07
max	999.000000	2.023042e+07	2.023052e+07

Figura 3.3: Estadísticos de Promociones

Las estadísticas descriptivas para las variables producto, idSecuenciaIni y idSecuenciaFin muestran un total de 26087 registros. La variable producto tiene valores que van desde 1 hasta 999. Las variables idSecuenciaIni y idSecuenciaFin van desde el 2011-01-04 hasta el 2023-05-17.

Análisis estadístico inicial de la tabla de **Stock** (figura 3.4):

	producto	idSecuencia	udsStock
count	707608.000000	7.076080e+05	707608.000000
mean	499.500000	2.021823e+07	127.028288
std	289.284031	6.478850e+03	131.422303
min	1.000000	2.021040e+07	-149.000000
25%	247.750000	2.021100e+07	69.000000
50%	499.500000	2.022040e+07	104.000000
75%	750.250000	2.022100e+07	144.000000
max	1000.000000	2.023040e+07	2460.000000

Figura 3.4: Estadísticos de Stock.

Las estadísticas descriptivas para las variables producto, idSecuencia y udsStock muestran un total de 707608 registros. La variable producto tiene entre 1 y 1000 productos. La variable udsStock muestra una media de 127.03 unidades en stock con una desviación estándar de 131.42, lo que sugiere una considerable variabilidad en los niveles de stock. El valor mínimo de udsStock es -149, lo que puede indicar **errores en los datos o ajustes negativos en el inventario**, y el valor máximo es 2460 unidades.

El siguiente paso es un análisis inicial sobre los ***outliers*** de las ventas y stock:

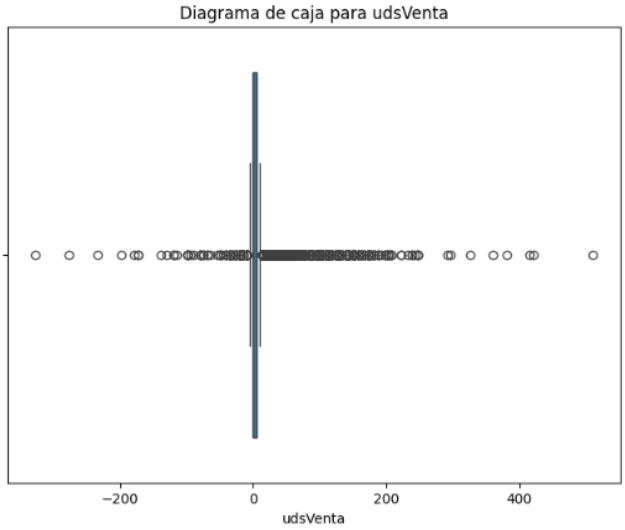

Figura 3.5: Diagrama de cajas de udsVenta.

El anterior diagrama de caja para **udsVenta** (figura 3.5) muestra que la mayoría de las ventas están concentradas alrededor de la mediana con un rango intercuartil muy estrecho. Sin embargo, hay una cantidad significativa de *outliers* en ambos extremos. Es importante poner el foco en las **unidades negativas**, ya que puede impactar en el modelo final.

El siguiente diagrama de caja para **udsStock** (figura 3.6), muestra que la mayoría de los niveles de stock están concentrados cerca de la mediana, con un rango intercuartil estrecho, al

igual que udsVenta. Sin embargo, hay una gran cantidad de *outliers* extendiéndose hasta 2500 unidades, indicando una alta variabilidad. También existen **unidades negativas**.

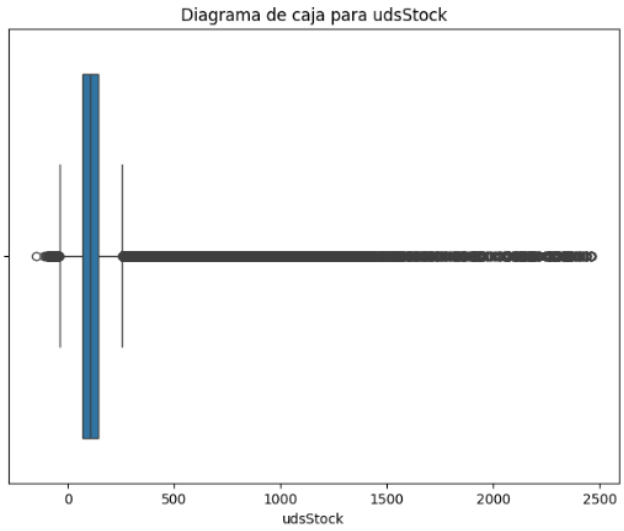

Figura 3.6: Diagrama de cajas de udsStock.

A continuación se muestra un gráfico de ventas históricas (figura 3.7) en el que hay una cantidad significativa de *outliers* en las ventas, con unidades que alcanzan valores altos y negativos. Las unidades vendidas negativas, representadas en rojo, indican **devoluciones o posibles errores** en los datos y están presentes en los datos **hasta el mes de octubre de 2022**.

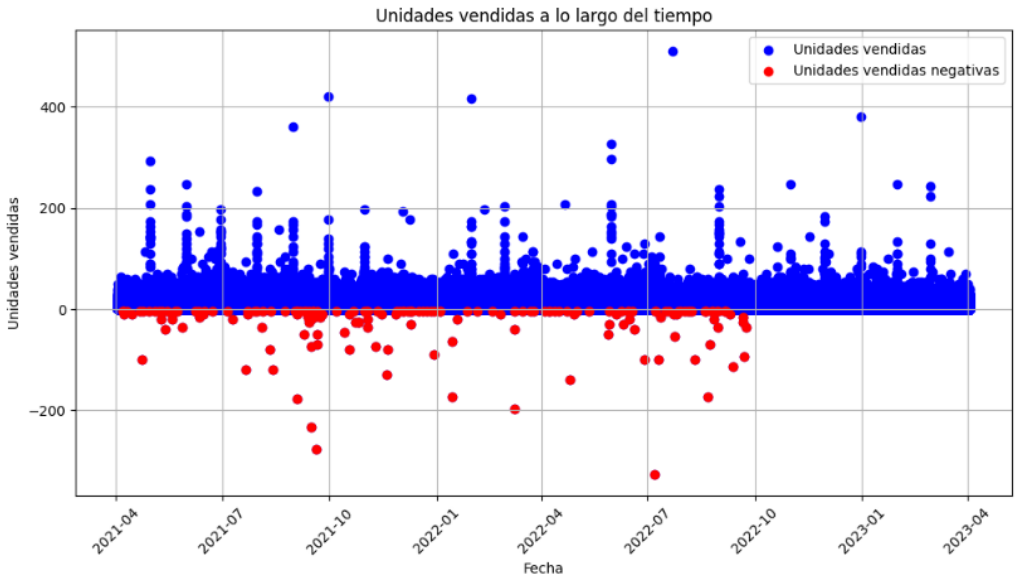

Figura 3.7: Ventas históricas.

En el siguiente gráfico (figura 3.8) se muestra, al igual que en las ventas, el histórico. Se puede observar que tiene una cantidad significativa de *outliers*. Las unidades en rojo son las negativas y pueden ser **posibles errores en los datos**. Están presentes en los **datos de 2021**.

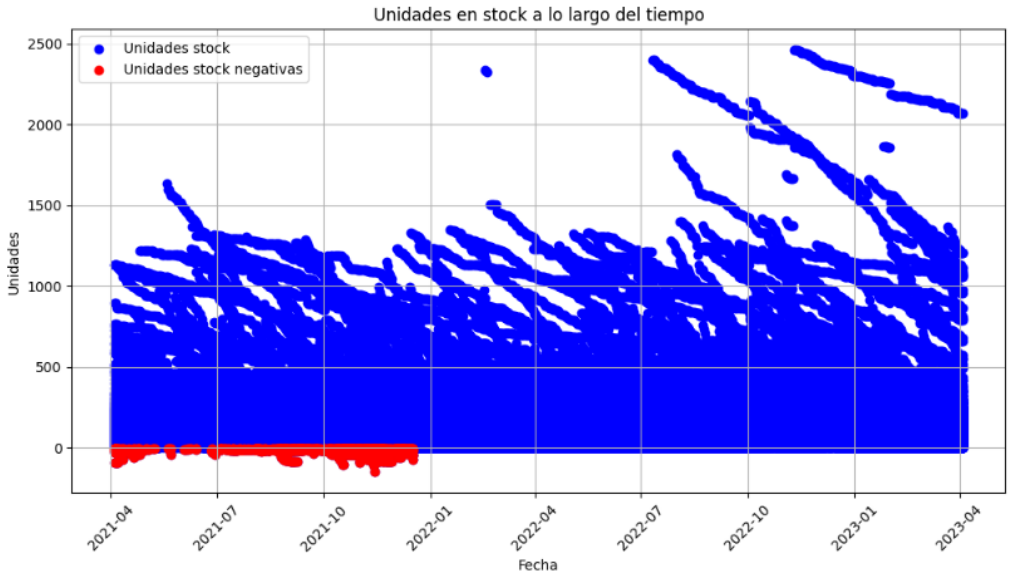

Figura 3.8: Stock histórico.

A continuación se muestran dos casos de unidades en venta y stock negativas (figuras 3.9 y 3.10). Se llega a la conclusión de que son **errores en el registro de los datos** (puede deberse a un malfuncionamiento de las máquinas que registran estos datos).

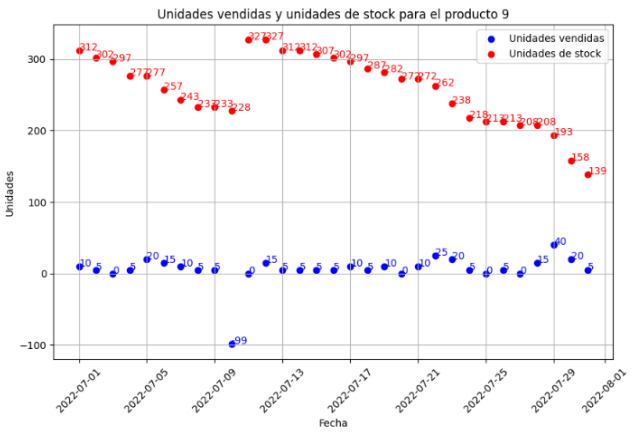

Figura 3.9: Ventas negativas del producto 9.

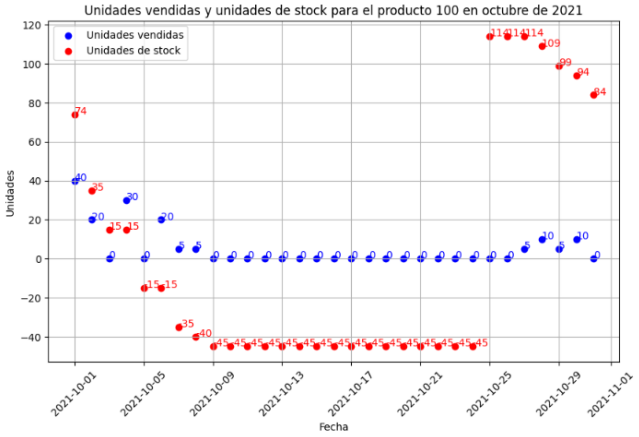

Figura 3.10: Stock negativo del producto 100.

3.2.3. Resumen de la calidad de los datos

Luego de hacer un primer primer acercamiento de los datos y haber visto impurezas (campos mal formateados, *outliers*, negativos...) se procede a modificar el conjunto de datos.

Además, se observa un fenómeno llamado rotura de stock, que sucede cuando no se producen ventas al no haber existencias. Esta situación se comprueba en varios periodos donde el stock y las ventas son cero. En la siguiente fase se tratan estos casos.

3.3. Preprocesado y análisis exploratorio de los datos

Durante esta fase se hace limpieza de posibles problemas encontrados en la anterior fase y se preparan los datos de manera que facilite el posterior análisis en profundidad a través de un EDA.

3.3.1. Limpieza y transformación de los datos

Se realizan una serie de modificaciones en los datos para eliminar posibles problemas y mejorar la calidad del dato:

1. **Cambio de tipos**: Se modifican los campos relacionados con las fechas para que sean de tipo *datetime*.

2. **Valores negativos en las udsVenta**: Se acuerda que, los casos negativos en ventas son debido a problemas electrónicos en el registro de las ventas, por lo que **se modifican a 0**.

3. **Valores negativos en las udsStock**: Se acuerda que, los casos negativos en stock son debido a incrongruencias en el registro de los datos, por lo que **se modifican a 0**.

4. **Valores a 0 en las ventas y stock (rotura de stock)**: Los casos de rotura de stock, se acuerda modificarlos con la **media de ventas del último mes para ese día de la semana y producto**.

5. **Transformación de la tabla de promociones**: Para añadir la información de las promociones a los registros de stock y ventas, se crea una *feature estaEnPromocion* que, en caso de que el producto esté en promoción el día del registro, equivale a 1, si no a 0.

3.3.2. Análisis de los datos

En la siguiente sección se analizan los datos de las ventas y stock, concretamente un análisis **multivariable temporal**:

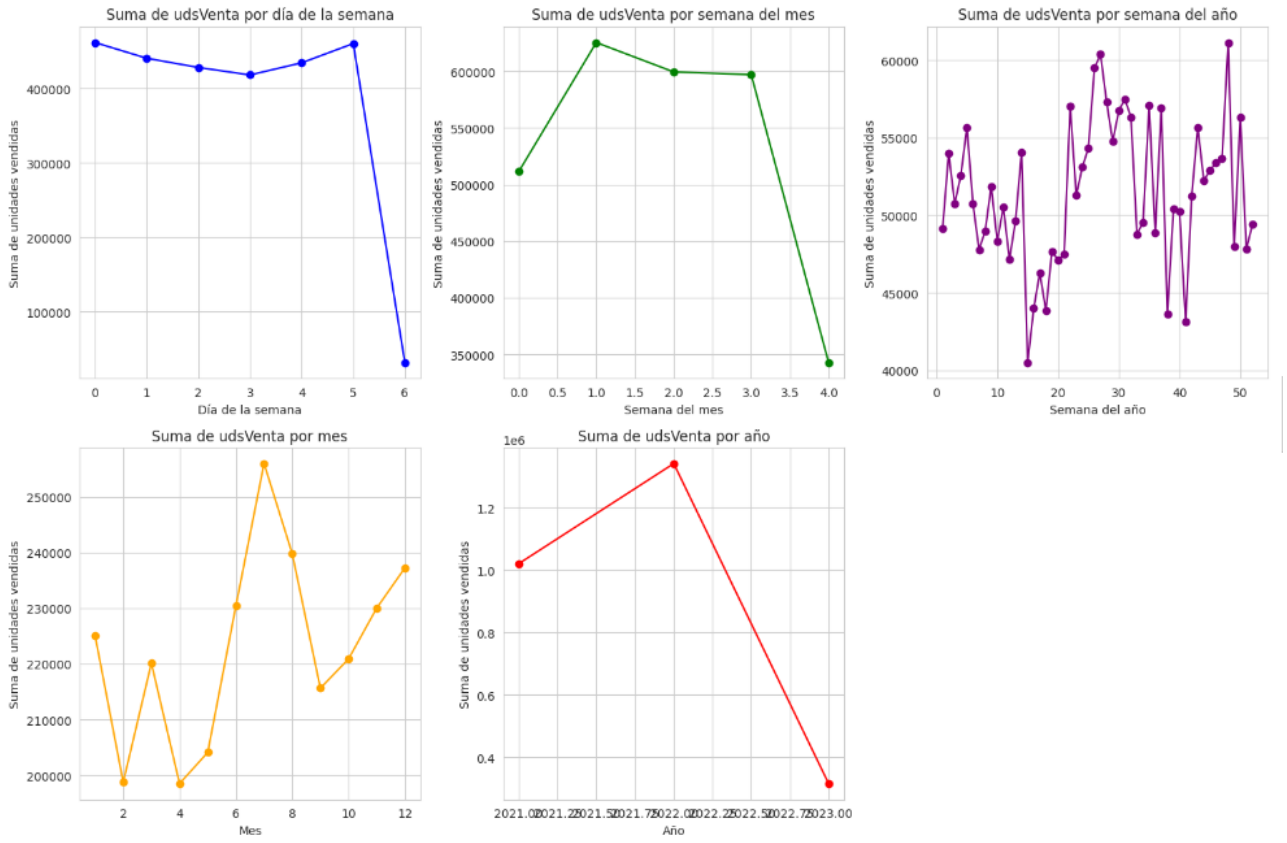

Figura 3.11: Ventas en diferentes agregaciones temporales.

En referencia a las gráficas (figura 3.11), se ha llegado a las siguientes conclusiones:

- **Ventas por día de la semana**: Las ventas son consistentemente altas de lunes a viernes, con un pico ligeramente más bajo el jueves. Las ventas caen drásticamente el domingo, indicando un posible cierre del punto de venta de de modo habitual.
- **Ventas por semana del mes**: Las ventas son más altas en las semanas centrales del mes, mientras que a principio y final de mes tienen caídas importantes.
- **Ventas por semana del año**: Las ventas muestran fluctuaciones semanales significativas a lo largo del año. Hay picos y valles a lo largo del año, sin embargo, en los meses centrales del año parece haber una clara subida y estabilidad de las ventas, coincidente con los meses de verano.

- **Ventas por mes**: Las ventas presentan picos en los meses en los meses de verano. Los meses de febrero y abril muestran las ventas más bajas. Hay una tendencia a tener variaciones significativas mes a mes.
- **Ventas por año**: El año 2022 es el que mayor ventas tiene, ya que también es el único año del que se tienen datos completos.

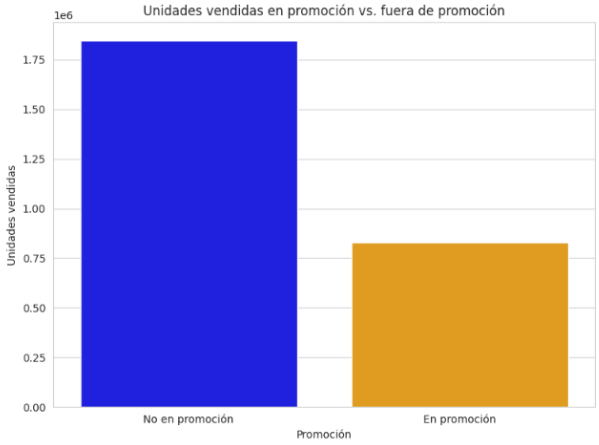

Figura 3.12: Ventas en promoción y no promoción.

En cuanto a las ventas producidas con promoción, se ve claramente que existen más ventas en temporadas sin promociones (figura 3.12) aunque se revisan los datos en la serie temporal siguiente.

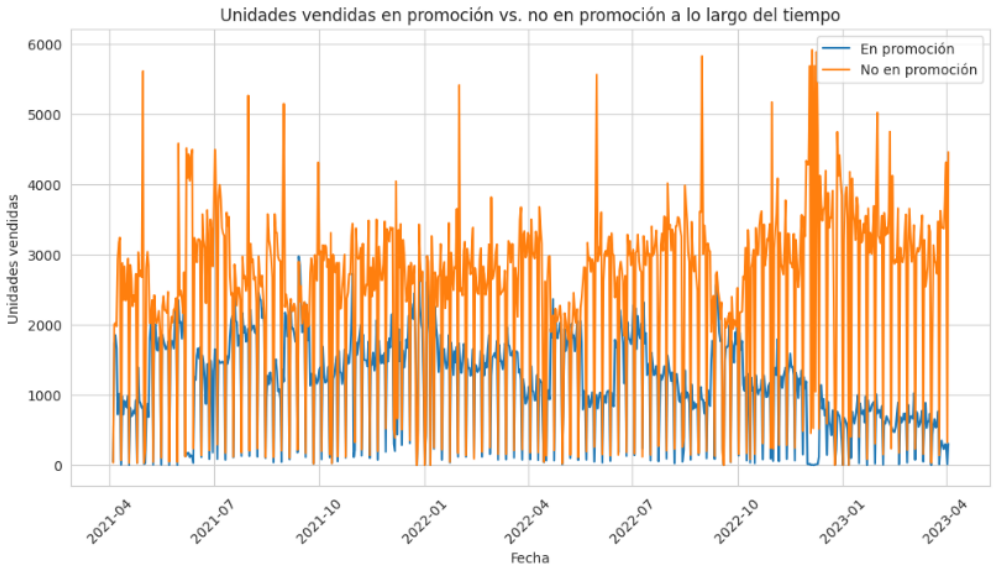

Figura 3.13: Ventas en campañas promocionales.

En la gráfica anterior (figura 3.13) se puede observar que las ventas de productos en promoción son consistentemente más bajas que las ventas de productos no en promoción. Hay ciertos periodos en los que las ventas de productos en promoción aumentan, coincidiendo con una disminución en las ventas de productos no en promoción. Esto puede indicar que en ciertos momentos, los consumidores están más inclinados a aprovechar las promociones disponibles.

En cuanto a las variables restantes, apertura/clausura y festivos, se observa un número de ventas pequeño en días en los que el punto de venta está cerrado y en días festivos, como se observa en las siguientes gráficas (figuras 3.14 y 3.15). Estas situaciones pueden estar sucediendo debido a errores en la tabla **Calendarios**, por lo que no se va realizar ninguna consideración a mayores.

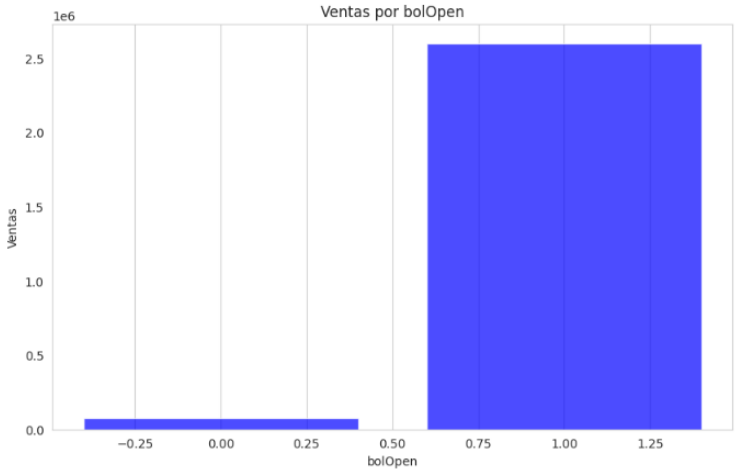

Figura 3.14: Ventas en apertura/clausura de punto de venta.

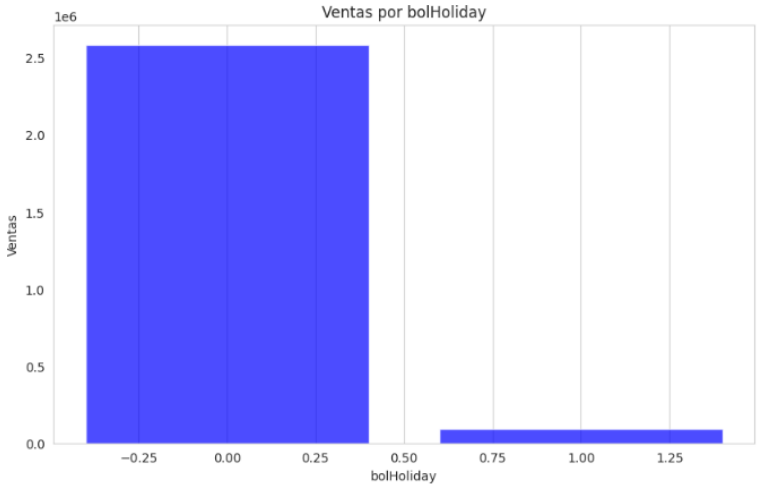

Figura 3.15: Ventas en festivos.

Es interesante saber el comportamiento de las ventas y el stock en la serie temporal. Por ello, se van a normalizar y reescalar los datos para ver las tendencias en cuanto a las ventas y el stock de los productos.

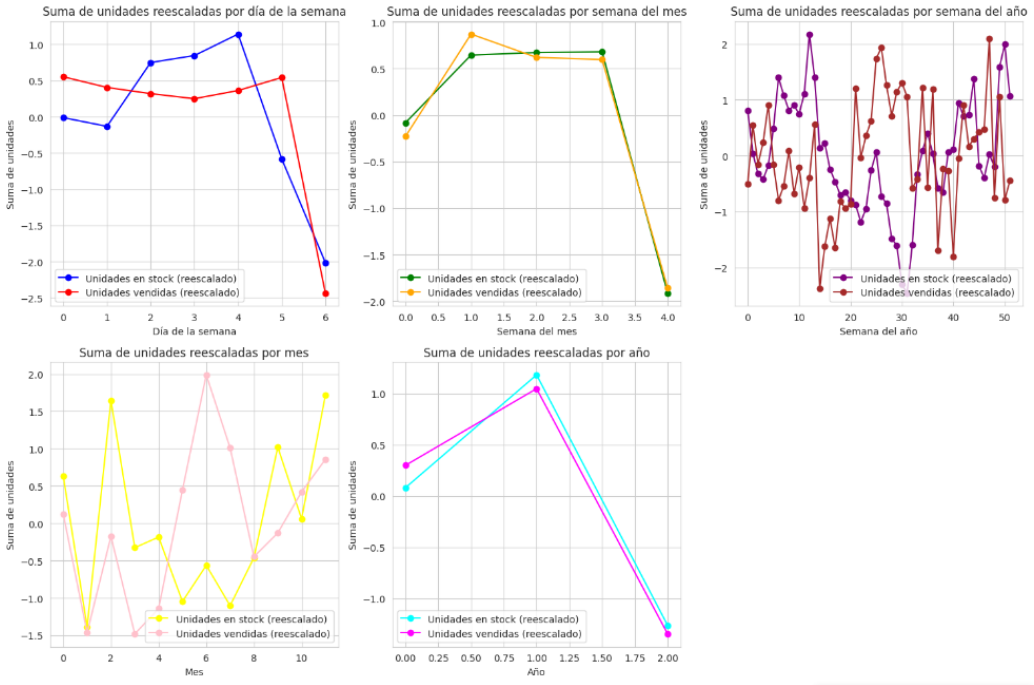

Figura 3.16: Ventas y stock.

Se puede observar en las gráficas (figura 3.16) que existe relación entre el **stock** y las **ventas**. El stock se ajusta regularmente en respuesta a las variaciones en las ventas a corto (semanal), medio (mensual) y largo plazo (anual). También existe cierta alineación entre ambos.

El siguiente paso es el de mostrar el **gráfico de correlaciones** entre las variables del conjunto de datos (figura 3.17). En él, se incluyen también las *features* creadas.

Se puede ver que las correlaciones más fuertes se corresponden con las variables temporales, algo previsible. Respecto a udsVenta, se puede observar que tiene unas ligeras correlaciones con producto, udsStock, bolOpen y bolHoliday. En cuanto a udsStock, se observa que tiene ligeras correlaciones con udsVenta.

Además, analizando los gráficos de autocorrelación y autocorrelación parcial (figura 3.18) se puede observar que existen varios picos significativos que permanecen positivos y van decayendo lentamente, lo que puede indicar una autocorrelación.

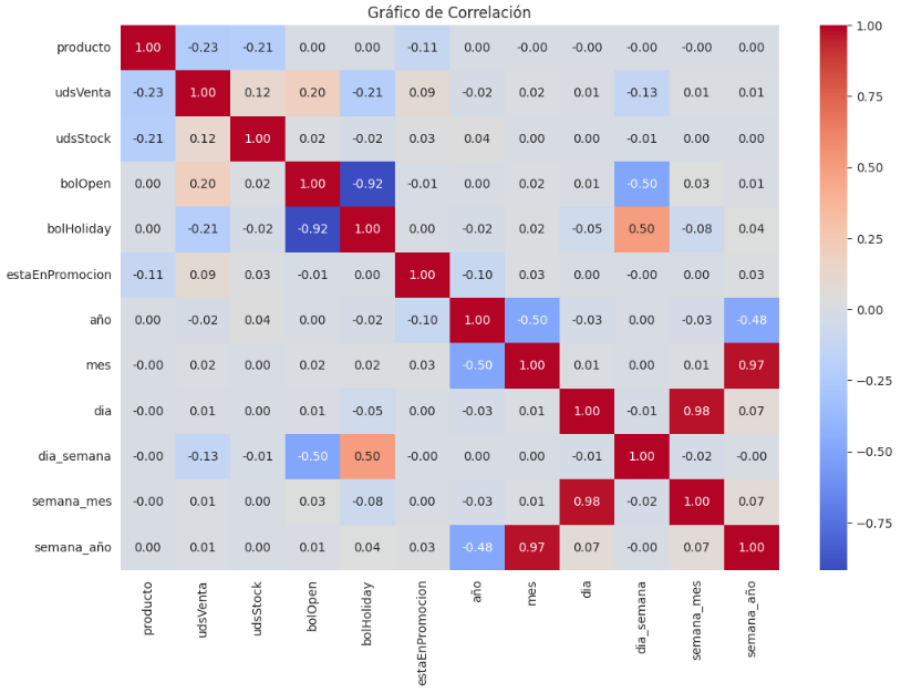

Figura 3.17: Correlación entre variables de datos de ventas.

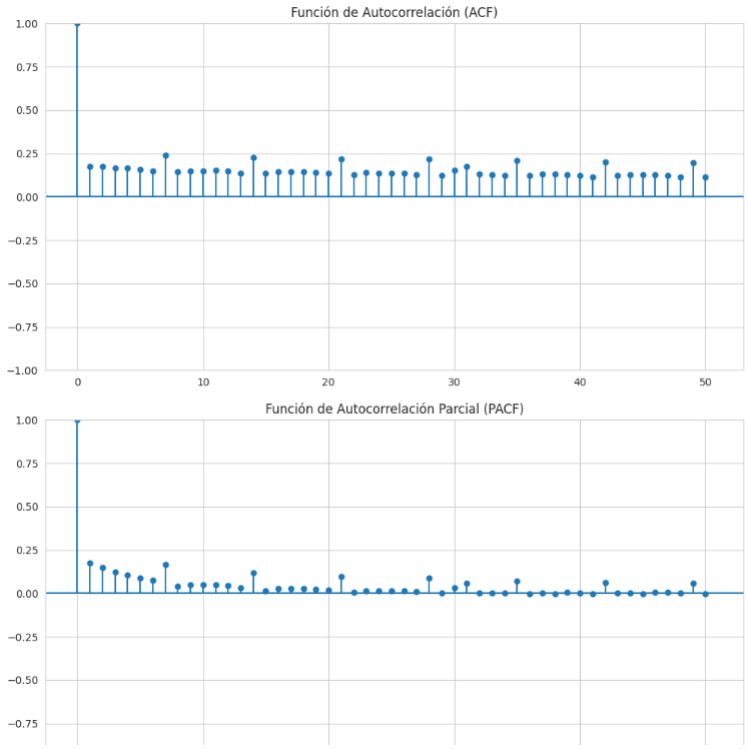

Figura 3.18: Gráficos de autocorrelación.

En cuanto al análisis temporal, es importante revisar si existe algún tipo de efecto **estacional**.

En primer lugar, se analizan los datos de ventas a lo largo del tiempo buscando algún tipo de tendencia.

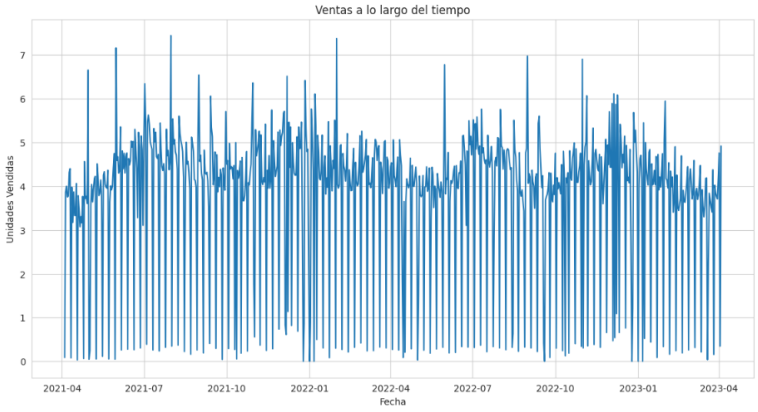

Figura 3.19: Gráfico de tendencia de las unidades vendidas.

Se puede observar en la gráfica anterior (figura 3.19) que las ventas pueden tener cierta tendencia estacional con picos y valles recurrentes cada año, lo que indica posibles fluctuaciones estacionales. Para intentar profundizar en el análisis, se descompone la serie temporal.

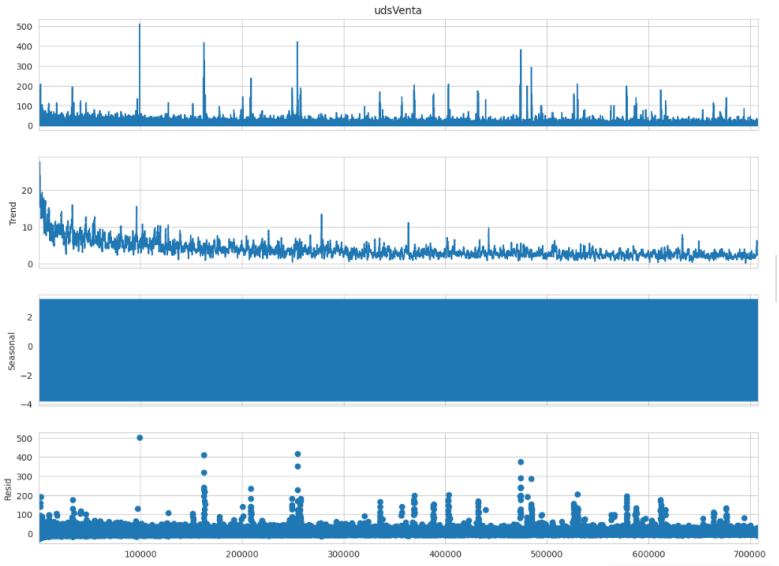

Figura 3.20: Descomposición de componentes de la serie temporal.

Al descomponer las componentes temporales (figura 3.20) no se observa que exista un patrón estacional claro y se observa en la gráfica de residuales que hay picos de *outliers* que generan ruido.

3.3.3. Escalado de los datos

En este proyecto se plantea el escalado de datos, pero realmente solo se utiliza en la etapa de *clustering*. Esta decisión es toma pensando en mantener los datos de ventas en sus unidades originales, ya que es importante para la interpretación clara de los resultados. Además, el proceso introduce una capa adicional de complejidad sin proporcionar beneficios significativos. En cambio, en la etapa de *clustering* sí que se utiliza para garantizar que todas las variables contribuyen equitativamente en la formación de *clusters*.

3.4. Feature Engineering

Con el fin de ayudar en la explicabilidad del modelo, se crean las siguientes *features*:

- **dia**: Representa el día del mes en que ocurrió la venta.
- **mes**: Indica el mes del año en que se realizó la venta.
- **año**: Especifica el año en que tuvo lugar la venta.
- **diaSemana**: Indica el día de la semana en que ocurrió la venta.
- **semanaMes**: Especifica la semana del mes en que se realizó la venta.
- **semanaAño**: Representa la semana del año en que ocurrió la venta.
- **estaEnPromocion**: Indica si el producto está en promoción en el momento de la venta (1 para sí, 0 para no).

3.5. Clustering de los datos

Buscar *clusters* en los datos es una fase muy importante en este proyecto. Mediante esta segmentación de datos, se buscan patrones o comportamientos en las ventas que ayuden a entender mejor el conjunto de datos y faciliten las predicciones.

Para la realización de esta fase se utiliza el algoritmo K-means, que agrupa datos similares mediante el centroide más cercano ajustándolos iterativamente. Además, se decide utilizar las variables **producto**, **unidades de venta** y **día de la semana** como variables a agrupar.

En primer lugar, se debe averiguar el número de agrupamientos necesarios. Para ello, se puede aplicar al regla del codo:

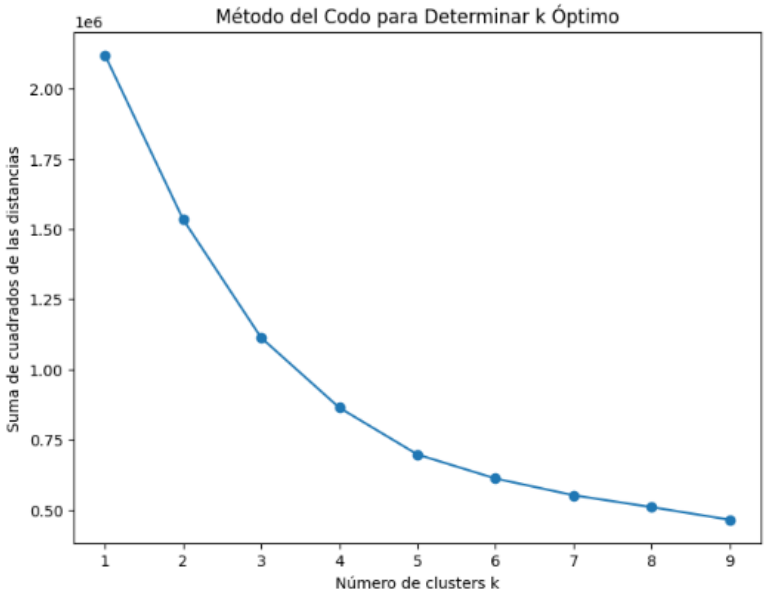

Figura 3.21: Gráfica con la regla del codo.

Según la gráfica anterior, (figura 3.21) el número óptimo de agrupaciones sería tres o cuatro, pero como existe suficiente variabilidad, se escogen **cuatro *clusters***.

A continuación se muestran las agrupaciones según las variables utilizadas (figura 3.22).

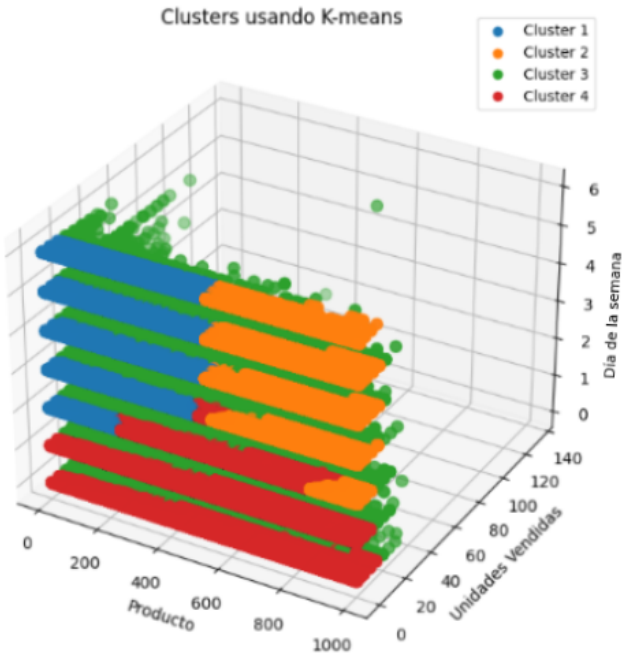

Figura 3.22: Gráfica con los *clusters* por producto, unidades vendidas y día de la semana.

Se puede observar que, según el día de la semana, determinados productos tienen cierto

patrón de ventas, pero el día de la semana no es lo suficientemente determinante como para ser la única característica en el modelo.

3.6. Modelado

En la siguiente sección se trata el tema de la selección de modelo, los indicadores de precisión a utilizar para evaluar los mismos, la defínción de los conjuntos de entrenamiento, pruebas y validación y la elección de hiperparámetros.

3.6.1. Selección de modelos

Durante el modelado, se llevan a cabo diversas pruebas con múltiples enfoques para identificar el mejor modelo. Inicialmente se prueban diferentes algoritmos sin tratar los *outliers*, sentando una primera línea base en cuanto al rendimiento. Posteriormente, se implementa un modelo sin *clustering* para evaluar si la falta de agrupamiento afectaba en el rendimiento. Luego se prueban diferentes configuraciones de *clusters*, ajustando la segmentación de datos. Finalmente se prueba un modelo con el tratamiento de *outliers* y roturas de stock.

Este último enfoque es el que mejor resultados ofrece, por lo que se opta por utilizar los diferentes algoritmos de machine learning con la modificación de hiperparámetros para encontrar el mejor rendimiento.

Las variables seleccionadas para el modelo fueron las siguientes:

- **Producto**
- **Fecha**
- **Unidades de venta**
- **Apertura**
- **Festivo**
- **Está en promoción**
- **Día de la semana**
- **Semana del mes**
- **Semana del año**
- **Cluster**

En cuanto a los algoritmos utilizados, se ha optado por utilizar de base aquellos que se han utilizado en la investigación del estado del arte, ya que son los que mejores resultados han dado. Además, se han tenido en cuenta las **limitaciones computacionales**.

- **Naive**: El algoritmo *naive* con la última observación [10] es un método simple y directo de realizar predicciones en series temporales. La premisa es que el valor más reciente de la serie temporal es el mejor predictor para el siguiente valor. Es útil en escenarios donde los datos son altamente autocorrelacionados.

- **Random Forest**: El algoritmo *Random Forest* [5] combina múltiples árboles de decisión. Cada árbol se entrena con un subconjunto diferente del conjunto de datos y una selección aleatoria de características. Las predicciones de los árboles individuales se combinan (promedio para regresión o votación mayoritaria para clasificación) para producir una predicción más robusta y reducir el sobreajuste.

- **Decision Tree**: El algoritmo *Decision Tree* [2] utiliza un árbol de decisiones para dividir iterativamente el conjunto de datos en subconjuntos homogéneos basados en la característica que proporciona la máxima ganancia de información o reducción de impureza. Cada nodo del árbol representa una característica, cada rama representa una decisión basada en esa característica, y cada hoja representa un resultado o clase.

- **LightGBM**: El algoritmo *Light Gradient Boosting Machine* [3] está basado en árboles de decisión que mejora iterativamente un modelo combinando varios modelos débiles para crear uno fuerte. Utiliza histogram-based algorithms y técnicas de reducción de datos para acelerar el entrenamiento y manejar grandes volúmenes de datos eficientemente.

- **XGBoost**: El algoritmo *Extreme Gradient Boosting* [6] es una implementación optimizada del algoritmo de boosting basada en árboles de decisión. Se centra en la velocidad y el rendimiento, utilizando técnicas como la paralelización y el uso eficiente de la memoria. Al igual que otros métodos de boosting, XGBoost construye modelos secuenciales que corrigen los errores de los modelos anteriores, mejorando gradualmente la precisión de las predicciones.

- **LSTM**: El algoritmo *Long Short-Term Memory* [4] es un tipo de red neuronal recurrente (RNN) diseñada para manejar dependencias a largo plazo en datos secuenciales. A diferencia de las RNN tradicionales, LSTM utiliza celdas de memoria que pueden retener y olvidar información de manera controlada a través de puertas (*input, forget, y output gates*).

3.6.2. Definición de indicadores de precisión

Para evaluar la precisión de los modelos predictivos, se utilizan los siguientes indicadores: el EVS (Puntaje de Varianza Explicada), el MSE (Error Cuadrático Medio) y el RMSE (Raíz del Error Cuadrático Medio), el más relevante de los tres ya que es el que se va a utilizar para el cálculo del impacto económico del algoritmo. A continuación se explican estos indicadores y su relevancia en el contexto del análisis.

- **EVS**: El Puntaje de Varianza Explicada (figura 3.23) calcula la relación entre la varianza de la diferencia entre los valores reales y los valores predichos y la varianza de los valores reales. El puntaje resultante varía entre -infinito y 1, donde un puntaje de 1 indica una coincidencia perfecta entre los valores reales y los predichos, y un puntaje de 0 indica que el modelo no es mejor que predecir la media de los valores reales [7].

$$\text{EVS} = 1 - \frac{\text{Var}(y - \hat{y})}{\text{Var}(y)}$$

Figura 3.23: Ecuación EVS.

  donde:

  - y es el valor observado.
  - $\hat{y}$ es el valor predicho.
  - Var es la varianza.

- **MSE**: El Error Cuadrático Medio (figura 3.24) mide la cantidad de error en los modelos estadísticos. Evalúa la diferencia cuadrática media entre los valores observados y los valores predichos. Cuando un modelo no tiene error, el MSE es igual a cero. A medida que el error del modelo aumenta, su valor también aumenta. El error cuadrático medio también se conoce como desviación cuadrática media (MSD, por sus siglas en inglés) [8].

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

Figura 3.24: Ecuación MSE.

  donde:

- n es el número de observaciones.
- y_i es el valor observado.
- $\hat{y}_i$ es el valor predicho.

- **RMSE**: La Raíz del Error Cuadrático Medio (figura 3.25) es la desviación estándar de los residuos (errores de predicción). Los residuos miden qué tan lejos están los puntos de datos de la línea de regresión; el RMSE mide qué tan dispersos están estos residuos. En otras palabras, indica qué tan concentrados están los datos alrededor de la línea de mejor ajuste [9].

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

Figura 3.25: Ecuación RMSE.

donde:

- n es el número de observaciones.
- y_i es el valor observado.
- $\hat{y}_i$ es el valor predicho.

3.6.3. Definición de periodos de entrenamiento y validación

Luego de realizar múltiples pruebas con conjuntos de prueba y validación en el rango de fechas de 2023, se decide finalmente que se realice con fechas del año 2021 y 2022 y antes del periodo navideño. Concretamente, el rango de fechas del **conjunto de entrenamiento es desde el 04-04-2021 hasta el 31-05-2022, el conjunto de validación desde el 01-06-2022 hasta el 31-08-2022 y el conjunto de pruebas desde el 01-09-2022 hasta el 01-12-2022**.

3.6.4. Elección de hiperparámetros

Para optimizar los modelos de aprendizaje automático, se lleva a cabo una exhaustiva búsqueda de hiperparámetros para cada uno de los algoritmos mencionados. Cabe destacar que, debido a limitaciones computacionales, **no se han llevado a cabo pruebas con rangos elevados de hiperparámetros**.

A continuación, se muestra la configuración de los hiperparámetros:

- **Random Forest**: Se configuran los siguientes hiperparámetros: **bootstrap: True** para utilizar muestras bootstrap y reducir la varianza, **maxDepth: 10** para limitar la profundidad de los árboles y evitar el sobreajuste, **minSamplesLeaf: 2** y **minSamplesSplit: 2** para controlar la regularización y las divisiones de los nodos, y **nEstimators: 150** para determinar el número de árboles en el bosque.
- **Decision Tree**: Se utilizan los hiperparámetros: **maxDepth: 2** para limitar la profundidad del árbol, **minSamplesLeaf: 1** para permitir nodos hoja con una sola muestra, y **minSamplesSplit: 2** para establecer el mínimo de muestras necesarias para dividir un nodo.
- **LightGBM**: Los hiperparámetros son: **learningRate: 0.1** para ajustar la velocidad de aprendizaje, **nEstimators: 200** para definir el número de árboles, y **numLeaves: 31** para especificar el número máximo de hojas por árbol.
- **XGBoost**: Se configuran los hiperparámetros: **gamma: 0** para no tener restricciones adicionales en la partición de nodos, **learningRate: 0.01** para un ajuste lento y preciso, **maxDepth: 5** para equilibrar la complejidad del modelo, y **nEstimators: 1000** para determinar el número de árboles.
- **LSTM**: Se utilizan los hiperparámetros por defecto.

3.7. Evaluación

En esta sección se llevan a cabo los entrenamientos del modelo con diferentes algoritmos y se clasifican los resultados.

3.7.1. Resultados de precisión obtenidos

A continuación se muestran los resultados obtenidos clasificados por métrica evaluada (cuadro 3.2):

	M1		
	EVS	**MSE**	**RMSE**
RF	0.664	10.203	3.194
DT	0.629	11.266	3.356
LGBM	0.667	10.117	3.180
XGB	0.665	10.165	3.188
LSTM	0.640	10.914	3.303

Cuadro 3.2: Comparación de métricas para diferentes algoritmos en el modelo final.

3.7.2. Comparación de los resultados obtenidos de los distintos modelos

En la siguiente tabla (cuadro 3.3) se pueden observar los resultados de dos modelos **M1** siendo el modelo en el que se trataron las roturas de stock y posibles *outliers* y **M2**, sin tratar. Se observa claramente que, al tratar los datos, se obtienen mejores predicciones.

	M1			M2		
	EVS	**MSE**	**RMSE**	**EVS**	**MSE**	**RMSE**
RF	0.664	10.203	3.194	0.558	17.903	4.231
DT	0.629	11.266	3.356	0.575	17.14	4.140
LGBM	0.667	10.117	3.180	0.608	15.838	3.979
XGB	0.665	10.165	3.188	0.612	15.674	3.959
LSTM	0.640	10.914	3.303	0.480	22.543	4.747

Cuadro 3.3: Comparación de métricas para diferentes algoritmos.

En cuanto al modelo final, en la gráfica de la figura 3.26 se observa en cuantos productos se ha utilizado un algoritmo u otro. También se ve una pequeña tabla en la figura 3.27 con los primeros y últimos productos, observando el RMSE y algoritmo utilizado.

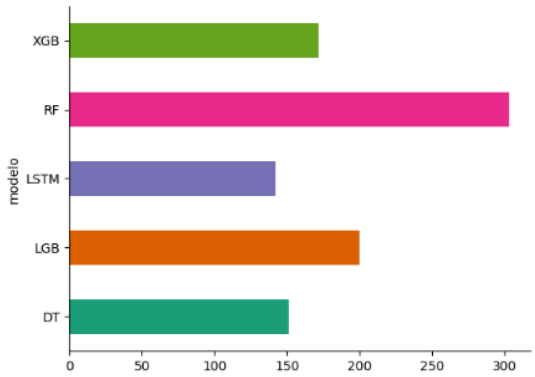

Figura 3.26: Combinación de algoritmos por producto.

	producto	RMSE	modelo
1	1	14.895816	RF
9	2	9.958897	LGB
10	3	8.772048	LGB
19	4	9.368164	LGB
23	5	8.199181	LSTM
...	...	...	...
4819	996	2.729071	LGB
4820	997	3.097152	LGB
4827	998	3.190243	DT
4834	999	4.453674	RF
4836	1000	2.974409	RF

968 rows × 3 columns

Figura 3.27: Tabla con algún producto, RMSE y algoritmo.

3.7.3. Conclusiones de los resultados obtenidos

En la comparación de los resultados, se evaluan las predicciones y su precisión utilizando el MSE, RMSE y EVS. Luego de considerar las ventajas y desventajas de cada uno de los modelos, se opta por una **estrategia de combinación de algoritmos por producto**, seleccionando para cada uno el modelo que menor RMSE obtubiese. Esta decisión se basa en el hecho de que el RMSE es la métrica necesaria para calcular estimar las unidades en stock y posterior impacto económico, por ello se requiere minimizar este valor para mejorar la precisión. Este enfoque presenta el mejor rendimiento y eficiencia económica.

3.8. Impacto de la mejora del algoritmo

En esta sección se analiza el impacto del modelo final. Para ello, se utilizan las ecuaciones recogidas por Yamazaki et al. [19], presentadas en este proyecto en la sección 2.3. Además, se comparan los resultados con un escenario en el que la empresa utiliza como método de referencia naive con la última observación conocida.

3.8.1. Impacto sobre los procesos y económico

Luego de obtener el RMSE asociado a la predicción de demanda de los productos, se pueden estimar las unidades en stock para cada producto. Para ello, en primer lugar se calcula la raíz del ciclo de aprovisionamiento (figura 3.28) (la raíz cuadrada de la suma de días entre pedido y el lead time).

	producto	diasEntrePedidos	diasLeadtime	raiz_ciclo_aprovisionamiento
0	1	14	15	5.385165
1	2	14	15	5.385165
2	3	14	15	5.385165
3	4	14	15	5.385165
4	5	14	15	5.385165
...	...	...	...	...
995	996	14	2	4.000000
996	997	14	2	4.000000
997	998	16	4	4.472136
998	999	7	2	3.000000
999	1000	14	2	4.000000

1000 rows × 4 columns

Figura 3.28: Tabla con el cálculo de la raíz del ciclo de aprovisionamiento.

Una vez obtenida la raíz del ciclo de aprovisionamiento, se pueden calcular las unidades de stock diarias a partir del producto de este valor, el RMSE y el factor de servicio (figura 3.29) (se calcula como la distribución normal estándar inversa para una probabilidad dada, en este caso 95 %).

	producto	diasEntrePedidos	diasLeadtime	raiz_ciclo_aprovisionamiento	RMSE	modelo	udsStock
0	1	14	15	5.385165	14.895816	RF	131.554933
1	2	14	15	5.385165	9.958897	LGB	87.953698
2	3	14	15	5.385165	8.772048	LGB	77.471837
3	4	14	15	5.385165	9.368164	LGB	82.736538
4	5	14	15	5.385165	8.199181	LSTM	72.412465
...	...	...	...	...	...	...	...
963	996	14	2	4.000000	2.729071	LGB	17.902707
964	997	14	2	4.000000	3.097152	LGB	20.317320
965	998	16	4	4.472136	3.190243	DT	23.398207
966	999	7	2	3.000000	4.453674	RF	21.912078
967	1000	14	2	4.000000	2.974409	RF	19.512125

968 rows × 7 columns

Figura 3.29: Tabla con las estimaciones diarias de unidades en stock.

El siguiente paso es el de estimar el coste anual, para ello en primer lugar obtenemos el

precio medio de cada producto (a partir de los datos iniciales) (figura 3.30).

	producto	eurPrecioMedio
0	1	68.730000
1	2	148.330000
2	3	169.000000
3	4	0.604383
4	5	4.553314
...	...	...
995	996	17.120000
996	997	17.710000
997	998	8.270000
998	999	8.270000
999	1000	40.730000

1000 rows × 2 columns

Figura 3.30: Tabla con el precio medio por producto.

Una vez obtenido el precio medio, se multiplica por las unidades en stock diarias de cada producto y por un valor que es el coste unitario porcentual (en este caso, un 5 %). De esta manera se puede estimar el coste anual por producto (figura 3.31).

	producto	eurPrecioMedio	diasEntrePedidos	diasLeadtime	raiz_ciclo_aprovisionamiento	RMSE	modelo	udsStock	CosteStock	CosteStock_365
0	1	68.730000	14	15	5.385165	14.895816	RF	131.554933	452.088527	165012.312526
1	2	148.330000	14	15	5.385165	9.958897	LGB	87.953698	652.308602	238092.639731
2	3	169.000000	14	15	5.385165	8.772048	LGB	77.471837	654.637019	238942.511781
3	4	0.604383	14	15	5.385165	9.368164	LGB	82.736538	2.500228	912.583086
4	5	4.553314	14	15	5.385165	8.199181	LSTM	72.412465	16.485836	6017.329970
...	...	...	...	...	...	...	...	...	...	...
963	996	17.120000	14	2	4.000000	2.729071	LGB	17.902707	15.324717	5593.521827
964	997	17.710000	14	2	4.000000	3.097152	LGB	20.317320	17.990987	6566.710261
965	998	8.270000	16	4	4.472136	3.190243	DT	23.398207	9.675159	3531.432939
966	999	8.270000	7	2	3.000000	4.453674	RF	21.912078	9.060644	3307.135203
967	1000	40.730000	14	2	4.000000	2.974409	RF	19.512125	39.736442	14503.801309

968 rows × 10 columns

Figura 3.31: Tabla con el coste anual por producto.

Esta forma de estimar el coste anual se puede aplicar a las predicciones del modelo de machine learning. También se aplica esta estimación de coste anual sobre un escenario en el que la empresa utilice **naive como previsión inicial**. La elección de **naive con el valor más reciente del histórico** es una práctica común debido a su simplicidad y capacidad para establecer una línea base mínima de rendimiento [10].

En el siguiente gráfico (figura 3.32) se compara el coste anual estimado para cada producto entre el modelo final y el modelo *naive*. Se observa que, en general, el modelo de machine learning presenta menores costes anuales en comparación con el *naive*, especialmente en los productos con mayores costes. Esta diferencia es significativa y se refleja en los costes anuales totales estimados: el modelo final tiene un coste anual total de **18,691,332 €**, mientras que el modelo *naive* estima un coste mucho mayor, de **36,099,255 €**, como se observa en la siguiente tabla (cuadro 3.4). La utilización de este algoritmo frente a una previsión inicial mediante *naive* supondría un ahorro económico de aproximadamente la **mitad** del prespuesto de stock.

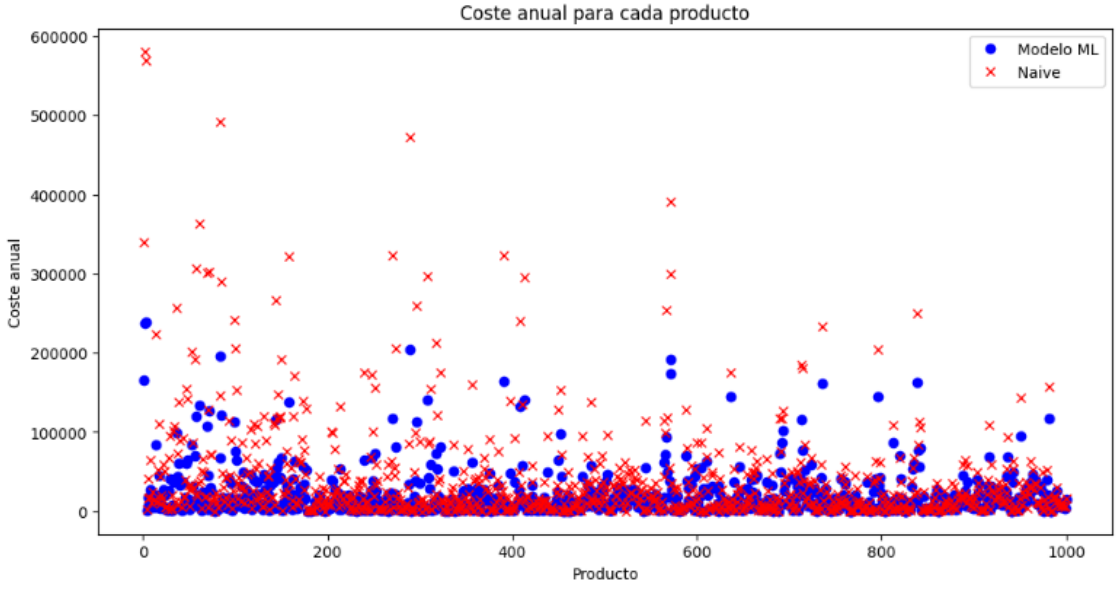

Figura 3.32: Gráfica con los costes anuales por producto.

Escenario óptimo (modelo de aprendizaje automático)	**Escenario inicial (modelo naive)**
18,691,332 €	36,099,255 €

Cuadro 3.4: Comparación de modelos de previsión

Esta situación se observa también cuando se calcula el ratio entre modelo final y naive por producto. En la siguiente gráfica (figura 3.33) se ve claramente que la previsión inicial tiende a estimar costes anuales significativamente más altos que la predicción de este proyecto.

Este patrón confirma que el modelo combinado de algoritmos de machine learning proporciona estimaciones de coste mucho más eficientes y consistentes, resultando en un ahorro económico considerable.

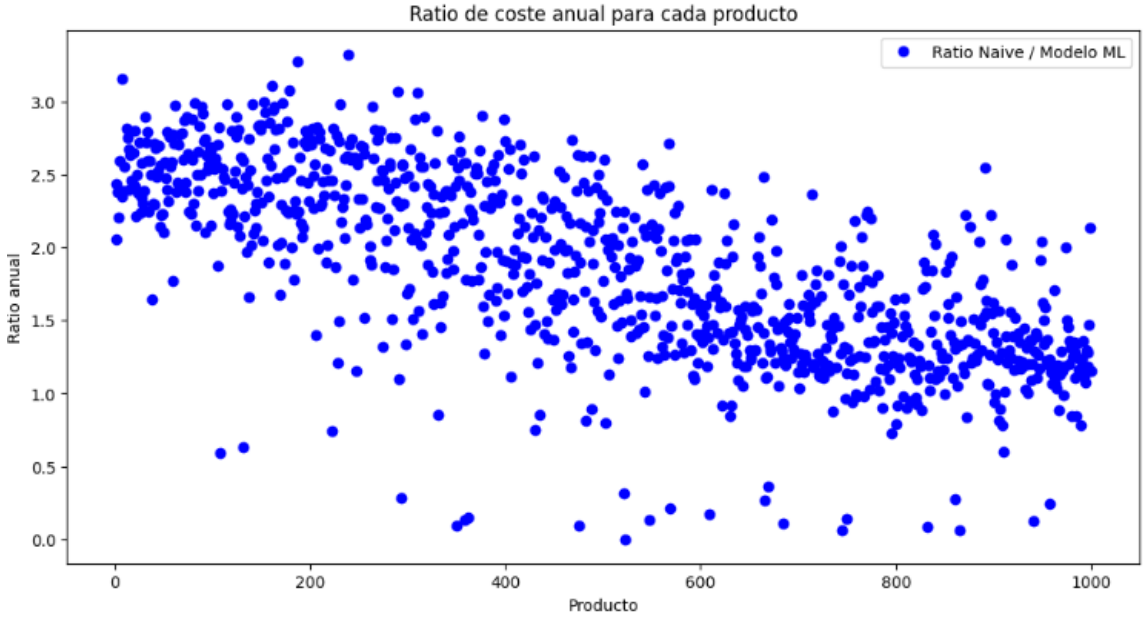

Figura 3.33: Gráfica con el ratio anual por producto.

Capítulo 4

Conclusiones

4.1. Lecciones aprendidas

El desarrollo de este proyecto ha sido una experiencia enriquecedora, especialmente considerando que se trata de un caso real con datos del mundo empresarial. Los datos del mundo real presentan problemas de calidad, como valores faltantes, inconsistencias y casos o situaciones que aumentan la complejidad de la problema a tratar.

Una de las lecciones aprendidas fue la importancia de la fase de preparación de los datos. Dedicar tiempo a la limpieza y transformación de los datos es crucial para realizar predicciones precisas. Un preprocesamiento adecuado es la base sobre la cual se construye un modelo predictivo óptimo.

Además, se ha comprendido la importancia de conocer los algoritmos utilizados, ya que cada uno tiene sus fortalezas y debilidades y por ello es esencial seleccionar el adecuado para el problema en cuestión. Mediante la investigación del estado del arte, se ha podido observar qué técnicas se llevan a cabo y tener una base con la que partir en este proyecto.

Otra lección importante es la necesidad de disponer de capacidad computacional para realizar las tareas de entrenamiento de manera eficaz. Esto se debe a la complejidad de los modelos utilizados, ya que si no se dispone del hardware adecuado, los entrenamientos se pueden prolongar horas y horas.

4.2. Logro de los objetivos

El proyecto ha logrado alcanzar los objetivos iniciales, que se centraban en la búsqueda de algoritmos que estimen con precisión la demanda necesaria y también mejorar el impacto económico. A través del uso de técnicas avanzadas de modelado y optimización del uso de algoritmos se ha podido mejorar la gestión de inventario y minimizar los costes asociados al

stock.

Sin embargo, a pesar de encontrar buenas estimaciones y minimizar el impacto económico, aún existe espacio para mejorar las estimaciones y alcanzar una mayor precisión en los modelos predictivos.

4.3. Seguimiento de la planificación y metodología

En el desarrollo de este proyecto se intentó seguir de manera rigurosa la planificación inicial. Sin embargo, hubieron fases del proyecto que requirieron más tiempo de lo previsto. Estas fueron la de limpieza de datos y modelado, ya que son tareas fundamentales para garantizar resultados óptimos y de calidad. Además, se empleó la metodología CRISP-DM que tiene un enfoque iterativo que ayuda a refinar continuamente el enfoque de los datos y mejorar los resultados progresivamente.

Cabe destacar que una de las limitaciones significativas fue la de la infraestructura disponible para los entrenamientos. Debido a la complejidad de los algoritmos utilizados, se requirieron múltiples horas de ejecución para completarlos. Fue una limitación influyente a la hora de tomar decisiones en cuanto a las diferentes pruebas con hiperparámetros y utilización de algoritmos de redes neuronales.

4.4. Líneas de trabajo futuro

En futuros trabajos, el objetivo principal será profundizar en mejorar la precisión y estimaciones de los modelos. Para ello, se marcan ciertas metas:

4.4.1. Mejorar la precisión del modelo

Se continuará experimentando con diferentes algoritmos y técnicas de modelado para encontrar los mejores resultados en términos de precisión, además de optimización de hiperparámetros.

4.4.2. Profundizar en las redes neuronales

Debido al potencial de las redes neuronales para capturar patrones complejos, se plantea explorar a fondo estas arquitecturas, incluyendo diferentes algoritmos. Se investigará a fondo cómo pueden ser aplicados y ajustados para este problema de predicción en concreto.

4.4.3. Mejorar la infraestructura

Se reconoce la necesidad de utilizar máquinas más potentes que soporten entrenamientos intensivos de forma eficiente. Para ello, la utilización de hardware como GPUs o TPUs será esencial para acelerar el proceso de entrenamiento y la experimentación con redes neuronales, además de la búsqueda de hiperparámetros.

4.4.4. Ampliación de datos

Una de las formas de mejorar las predicciones es contar con un mayor histórico de datos. Se planea expandir el conjunto de datos histórico y capturar una mayor variabilidad. Además, se podrían analizar las tendencias a largo plazo. También sería interesante integrar más datos empresariales que puedan influir en las predicciones. Cualquier información relevante puede enriquecer el contexto y mejorar la capacidad de predicción del modelo.

Glosario

Agenda 2030 Un plan de acción adoptado por todos los Estados Miembros de las Naciones Unidas en 2015, que incluye 17 Objetivos de Desarrollo Sostenible (ODS) destinados a poner fin a la pobreza, proteger el planeta y garantizar la paz y la prosperidad para todas las personas. 3

EDA Análisis Exploratorio de Datos, un enfoque para analizar conjuntos de datos para resumir sus principales características a menudo utilizando métodos visuales. 2, 3, 25

git Un sistema de control de versiones distribuido diseñado para manejar todo tipo de proyectos con rapidez y eficiencia. 3

IDE Entorno de Desarrollo Integrado, una aplicación de software que proporciona herramientas completas para el desarrollo de software, incluyendo un editor de código, un depurador, un compilador y, a menudo, un entorno gráfico de usuario. 3, 4

rotura de stock Una situación en la que no hay suficiente inventario disponible para satisfacer la demanda de los clientes, lo que puede resultar en pérdidas de ventas y disminución de la satisfacción del cliente. 1, 2, 4, 9, 25, 34, 39

stacking Una técnica de ensamblado en aprendizaje automático que combina múltiples modelos predictivos a través de un meta-modelo, el cual se entrena para hacer predicciones a partir de las predicciones de los modelos base. 11, 13

stock de seguridad Cantidad adicional de inventario que se mantiene para reducir el riesgo de escasez de stock debido a variaciones en la demanda o en el tiempo de entrega. 16

Bibliografía

[1] Competencia de compromiso ético y global (cceg) y objetivos de desarrollo sostenible (ods), 2024. Fecha de acceso: 08-03-2024. URL: https://www.uoc.edu/portal/es/compromis-social/index.html.

[2] Definición del algoritmo decision tree, 2024. Fecha de acceso: 20-05-2024. URL: https://towardsdatascience.com/decision-tree-in-machine-learning-e380942a4c96.

[3] Definición del algoritmo lightgbm, 2024. Fecha de acceso: 20-05-2024. URL: https://towardsdatascience.com/a-quick-guide-to-lightgbm-library-ef5385db8d10.

[4] Definición del algoritmo lstm, 2024. Fecha de acceso: 20-05-2024. URL: https://medium.com/analytics-vidhya/introduction-to-long-short-term-memory-lstm-a8052cd0d4cd.

[5] Definición del algoritmo random forest, 2024. Fecha de acceso: 20-05-2024. URL: https://towardsdatascience.com/random-forests-algorithm-explained-with-a-real-life-example-and-some-python-code-

[6] Definición del algoritmo xgboost, 2024. Fecha de acceso: 20-05-2024. URL: https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/.

[7] Definición del evs, 2024. Fecha de acceso: 17-05-2024. URL: https://permetrics.readthedocs.io/en/v1.4.3/pages/regression/EVS.html.

[8] Definición del mse, 2024. Fecha de acceso: 17-05-2024. URL: https://statisticsbyjim.com/regression/mean-squared-error-mse/.

[9] Definición del rmse, 2024. Fecha de acceso: 17-05-2024. URL: https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/.

[10] Definición y aplicación de algoritmos naive para las previsiones, 2024. Fecha de acceso: 20-05-2024. URL: https://otexts.com/fpp2/simple-methods.html.

[11] Documentación sobre deepar, 2024. Fecha de acceso: 24-03-2024. URL: https://docs.aws.amazon.com/sagemaker/latest/dg/deepar.html.

[12] Gráfico con la metodología crisp-dm. 2024. Fecha de acceso: 03-03-2024. URL: https://healthdataminer.com/wp-content/uploads/2019/11/800px-CRISP-DM_Process_Diagram.png.

[13] Metodología crisp-dm, 2024. Fecha de acceso: 03-03-2024. URL: https://healthdataminer.com/data-mining/crisp-dm-una-metodologia-para-mineria-de-datos-en-salud/.

[14] Adnan et al. Effective demand forecasting model using business intelligence empowered with machine learning. 2020. Fecha de acceso: 24-03-2024. URL: https://www.researchgate.net/publication/342326031_Effective_Demand_Forecasting_Model_Using_Business_Intelligence_Empowered_With_Machine_Learning.

[15] Ariful Islam Arif et al. Comparison study: Product demand forecasting with machine learning for shop. 2019. Fecha de acceso: 25-03-2024. URL: https://ieeexplore.ieee.org/abstract/document/9117395.

[16] Real Carbonneau et al. Application of machine learning techniques for supply chain demand forecasting. 2008. Fecha de acceso: 24-03-2024. URL: https://www.sciencedirect.com/science/article/pii/S0377221706012057.

[17] Takashi Tanizakia et al. Demand forecasting in restaurants using machine learning and statistical analysis. 2018. Fecha de acceso: 24-03-2024. URL: https://www.sciencedirect.com/science/article/pii/S2212827119301568.

[18] Vinit Taparia et al. Improved demand forecastingof a retail store using a hybridmachine learning model. 2023. Fecha de acceso: 18-03-2024. URL: https://www.researchgate.net/publication/375777049_Improved_Demand_Forecasting_of_a_Retail_Store_Using_a_Hybrid_Machine_Learning_Model.

[19] Keisuke Shida y Takashi Kanazawa TTomoaki Yamazaki. An approach to establishing a method for calculating inventory. 2019. Fecha de acceso: 25-03-2024. URL: https://www.tandfonline.com/doi/full/10.1080/00207543.2015.1076179.

[20] I-Fei Chen y Chi-Jie Lu. Demand forecasting for multichannel fashion retailers by integrating clustering and machine learning algorithms. 2021. Fecha de acceso: 25-03-2024. URL: https://www.mdpi.com/2227-9717/9/9/1578.

[21] Jakob Huber y Heiner Stuckenschmidt. Daily retail demand forecasting using machine learning with emphasis on calendric special days. 2020. Fecha de acceso: 16-03-2024. URL: https://www.sciencedirect.com/science/article/pii/S0169207020300224.

[22] Shuojiang Xu y Hing Kai Chanb. Forecasting medical device demand with online search queries: A big data and machine learning approach. 2019. Fecha de acceso: 25-03-2024. URL: https://www.sciencedirect.com/science/article/pii/S2351978920302699.

[23] Sai Ramya y K. Vedavath. An advanced sales forecasting using machine learning algorithm. 2020. Fecha de acceso: 22-03-2024. URL: https://www.ijisrt.com/assets/upload/files/IJISRT20MAY134.pdf.

[24] F. Robert Jacobs y Richard B. Chase. Administración de operaciones. producción y cadena de suministros. 2014. Fecha de acceso: 16-03-2024. URL: https://ucreanop.com/wp-content/uploads/2020/08/Administracion-de-Operaciones-Produccion-y-Cadena-de-Suministro-13edi-Chase.pdf.

[25] Resul Tugay y Sule Gunduz Oguducu. Demand prediction using machine learning methods and stacked generalization. 2009. Fecha de acceso: 24-03-2024. URL: https://arxiv.org/abs/2009.09756.

Apéndice A

Código fuente

Repositorio git en github con el código fuente del proyecto