******

**Análisis de inventario**

Unamuno Acha, Isabel

2023 - 2024

Máster Universitario en

Análisis de Datos para la Inteligencia de Negocio / Business Analytics

TRABAJO DE FIN DE MÁSTER

**ANÁLISIS DE INVENTARIO**

TRABAJO DE FIN DE MÁSTER PRESENTADO EN: Mondragon Unibertsitatea

PARA LA OBTENCIÓN DEL TÍTULO DE: Máster Universitario en Análisis de Datos para la Inteligencia de Negocio / Business Analytics

AUTOR/A: Isabel Unamuno Acha

DIRECTOR/A: XXX

TUTOR/A: Jon Perez Visaires

ORGANIZACIÓN EN LA QUE HA REALIZADO EL PROYECTO: Datua

FECHA DE DEFENSA: Bilbao, a DD de MM de AAAA

El autor/la autora del Trabajo de Fin de Máster, autoriza a la Facultad de Empresariales de Mondragon Unibertsitatea, con carácter gratuito y con fines exclusivamente de investigación y docencia, los derechos de reproducción y comunicación pública de este documento siempre que: se cite el autor/la autora original, y el uso que se haga de la obra no sea comercial.

****

**Reconocimiento – NoComercial – CompartirIgual (by-nc-sa):** No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

DECLARACIÓN DE ORIGINALIDAD

Yo Isabel Unamuno Acha

Declaro que este Trabajo de Fin de Máster es original, fruto de mi trabajo personal, y que no ha sido previamente presentado para obtener otro título o calificación profesional.

Las ideas, formulaciones, imágenes, ilustraciones tomadas de fuentes ajenas han sido debidamente citadas y referenciadas.

**RESUMEN**

**LABURPENA**

**ABSTRACT**

**ÍNDICE**

RESUMEN 3

LABURPENA 4

ABSTRACT 5

1. INTRODUCCIÓN 9

1.1. Presentación de la empresa y problemática 9

1.2. Marco teórico 9

1.2.1. Red Neuronal Artificial multicapa con Embeddings 10

1.2.2. Gradient Boosting Regressor 10

1.2.3. Decision Tree Regressor 10

1.2.4. Light Gradient Boosting Machine Regressor 10

1.3. Objetivos 13

1.4. Descripción de la metodología empleada 14

1.5. Planificación 14

2. DESARROLLO 15

2.1. Análisis de los datos 15

2.1.1. Entrada de datos: 15

2.1.2. Preparación de datos 15

2.1.3. Limpieza y procesamiento 16

2.1.3.1. Tratamiento de Missings 16

2.1.3.2. Tratamiento de duplicados 16

2.1.3.3. Irregularidades en los datos 17

2.2. Modelo semántico en powerbi: 18

2.3. Informe PowerBI 22

2.4. Aplicación de Modelos para la predicción de la demanda 22

3. RESULTADOS 27

4. CONCLUSIONES Y DISCUSIÓN 28

**ÍNDICE DE FIGURAS**

Figura 1: Modelo estrella 18

Figura 2:Serie temporal de ventas y compras 24

Figura 3: Ventas diarias 26

Figura 4: Ventas diarias con y sin estacionalidad 27

Figura 5: Predicción de ventas sin estacionalidad 28

**ÍNDICE DE TABLAS**

Tabla 1: Descripción de las tablas 20

# INTRODUCCIÓN

## Presentación de la empresa y problemática

La empresa es una empresa mayorista de vinos y licores que opera en múltiples ubicaciones de Estados Unidos. Cuenta aproximadamente con 80 establecimientos y unas ventas totales superiores a 450 millones de dólares. En los últimos años, la empresa ha tenido problemas con la gestión del inventario, por lo que el equipo directivo quiere analizarlo para identificar áreas de mejora y optimizar sus prácticas de gestión de inventario. Esta empresa gestiona ventas y costes de bienes vendidos que alcanzan los cientos de millones, lo que hace que las hojas de cálculo tradicionales resulten insuficientes. El vasto volumen de datos, que incluye millones de registros de ventas, compras e inventario, requiere un enfoque sofisticado para un análisis eficaz. Por ello se quiere conseguir diseñar un cuadro de mando que permita a simple vista tomar decisiones de valor.

## Marco teórico

En el entorno empresarial actual, la aplicación de algoritmos de Machine Learning para la gestión de inventarios se ha convertido en una herramienta esencial. Estas técnicas avanzadas permiten a las empresas predecir con mayor precisión la demanda futura y ajustar sus niveles de stock de manera eficiente. Al optimizar los costos de almacenamiento y reducir las pérdidas asociadas con la escasez de productos, las empresas pueden no solo mejorar su rentabilidad, sino también elevar la satisfacción del cliente al asegurar la disponibilidad continua de los productos más demandados. La implementación de algoritmos de Machine Learning en la gestión de inventarios proporciona una ventaja competitiva significativa. Estas herramientas permiten a las empresas operar de forma más eficiente, responder mejor a las fluctuaciones del mercado y adaptarse rápidamente a las necesidades cambiantes de los consumidores. Por ello, los beneficios, en el siguiente apartado, primero de todo se ha resumido el concepto de machine learning y a continuación, se han analizado cuáles son los diferentes algoritmos de machine learning que se emplean actualmente para la predicción de la demanda y la mejora en la gestión de stock.[1]

Los algoritmos de Machine Learning permiten identificar patrones complejos en grandes volúmenes de datos, infiriendo sus propias reglas para reconocer patrones similares en nuevos conjuntos de datos. Esto da lugar a la creación de sistemas inteligentes que mejoran de forma autónoma mediante la observación de datos. Estos sistemas pueden aprender a predecir comportamientos, detectar similitudes o anomalías de manera automática, y tomar decisiones adecuadas basadas en los datos analizados. Los modelos de Machine Learning analizan datos reales que representan el proceso que se desea mejorar o automatizar. Este proceso, conocido como entrenamiento de algoritmos, permite que el sistema extraiga conclusiones relevantes y aprenda a realizar tareas sin necesidad de programación previa, simplemente observando y analizando los datos proporcionados. En los proyectos de Machine learning, hay tres pasos que se deben seguir, que son los siguientes. Se comienza con la revisión de datos, el cual implica examinar los datos disponibles para determinar cuáles son útiles, asegurando su correcto almacenamiento y formatos adecuados. Es crucial limpiar los datos eliminando registros antiguos, incompletos o erróneos para garantizar la calidad del dataset. Se sigue con la organización de datos, siendo esta fundamental para facilitar el aprendizaje automático. Se seleccionan datos que reflejen las decisiones que el sistema deberá tomar automáticamente, estructurándolos de manera que optimicen el proceso de entrenamiento del modelo. Y por último, el entrenamiento y la validación del modelo. Durante esta fase, el modelo analiza los datos para identificar patrones y relaciones, ajustando sus parámetros para poder detectar estas características automáticamente en futuros datos. Este proceso permite al modelo configurar su propio sistema de detección basado en la información observada.[2]

Tras conocer los conceptos generales del Machine Learning, se ha visto como se aplican tanto las Redes neuronales como los modelos de Machine Learning en la gestión de inventarios actualmente. Las redes neuronales son ampliamente utilizadas para predecir el comportamiento del inventario. Los modelos de backpropagation (BPNN) destacan por su estructura simple y capacidad de aprendizaje. Entre los científicos que han desarrollado varios artículos, se han mencionado algunos que se han considerado interesantes. Šustrová desarrolló varios modelos de redes neuronales para optimizar la cantidad de stock, seleccionando el modelo con menor error cuadrático medio (MSE) y un coeficiente de determinación (R2) cercano a 1. Su modelo final fue una red feed-forward con backpropagation, función de entrenamiento TRAINGDX y función de transferencia TANSIG [3]. Shoujing Zhang combinó BPNN con clustering mejorado y PCA para gestionar el inventario de piezas de repuesto en mantenimiento de camiones, evitando el impacto subjetivo de los gestores de inventario[4]. Por otro lado, Praveen K.B. y otros utilizaron el modelo de regresión XGBoost para predicciones de demanda en pequeñas y medianas empresas, logrando reducir el stock y el capital dedicado[5]. Con la información adquirida en los artículos, se han recogido algunos de los algoritmos que se han considerado interesantes y se ha explicado la función de ellas brevemente.

### Red Neuronal Artificial multicapa con Embeddings

Los Embeddings son una técnica para convertir variables categóricas en representaciones continuas, que se utilizan como entradas en un modelo de red neuronal para aprendizaje supervisado. Estas representaciones permiten que las categorías similares se ubiquen más cerca entre sí y se ajustan durante el entrenamiento para minimizar la pérdida del modelo. Al emplear Embeddings, se pueden obtener predicciones más precisas en comparación con el uso exclusivo de las unidades de stock. En este proyecto, además de los Embeddings y las unidades de stock, también se considerarán las unidades de venta para mejorar la precisión de las predicciones[1].

### Gradient Boosting Regressor

Este estimador construye un modelo aditivo por etapas y permite optimizar funciones de pérdida diferenciables arbitrarias. En cada etapa se ajusta un árbol de regresión sobre el gradiente negativo de la función de pérdida dada[6].

### Random Forest Regressor

Un bosque aleatorio es un metaestimador que ajusta una serie de árboles de decisión regresores a varias submuestras del conjunto de datos y utiliza el promedio para mejorar la precisión predictiva y controlar el sobreajuste. Los árboles del bosque utilizan la mejor estrategia de división, lo que equivale a pasar splitter=«best» al DecisionTreeRegressor subyacente.

### Decision Tree Regressor

Un árbol de decisión es un tipo de algoritmo de aprendizaje supervisado que se usa comúnmente en el aprendizaje automático para modelar y predecir resultados basados en datos de entrada. Es una estructura en forma de árbol donde cada nodo interno prueba el atributo, cada rama corresponde al valor del atributo y cada noso hoja representa la decisión o predicción final. El algoritmo del árbol de decisión entra en la categoría de aprendizaje supervisado. Se pueden utilizar para resolver problemas tanto de regresión como de clasificación[7].

### Light Gradient Boosting Machine Regressor

Es un marco de código abierto ampliamente utilizado para el refuerzo por gradiente. Destacado por su capacidad para manejar conjuntos de datos de gran tamaño, ofrece un rendimiento superior en comparación con otros marcos de refuerzo por gradiente como XGBoost y CatBoost.Una de las características distintivas de Light GBM es su método de muestreo unilateral basado en el gradiente para dividir los árboles. Este enfoque reduce significativamente el uso de memoria y mejora la precisión del modelo. Además, Light GBM emplea un crecimiento por hojas en lugar de crecimiento por niveles, lo que lo hace considerablemente más rápido que los métodos tradicionales de crecimiento en profundidad[8].

Volviendo al entorno empresarial, vivimos en un momento de rápidos cambios en el área tecnológica, donde todas las empresas deben poder reaccionar rápidamente. Cualquier organización necesita adaptarse cada vez más rápido a un mundo en constante cambio. Además, cada día aumenta la complejidad de la cadena de suministro. Por lo que no hay otra manera de que las empresas sobrevivan y se adapten rápidamente a esos cambios que automatizando procesos en su gestión de suministros. Y convertir procesos manuales en procesos en los que las máquinas son supervisadas por humanos. Si bien no se pueden resolver todos los desafíos simplemente añadiendo tecnología, una estrategia de cadena de suministro moderna requiere una pila de tecnología actualizada. Las cadenas de suministro, tradicionalmente lineales y predecibles, enfrentan una importante fragmentación de la demanda debido al crecimiento del comercio digital y nuevos modelos de cumplimiento. Esta evolución está haciendo que las cadenas actuales se vuelvan obsoletas. Aunque nadie puede prever con certeza el futuro de la cadena de suministro, es claro que la inteligencia artificial jugará un papel fundamental para impulsar su resiliencia. Se espera una mayor visibilidad de extremo a extremo con sistemas más interconectados y un mayor uso de la IA y el machine learning para la previsión de la demanda y prácticas más sostenibles, respondiendo a las demandas de los consumidores. En un futuro cercano, las cadenas de suministro podrían ser altamente autónomas, con sistemas impulsados por IA que gestionen la mayoría de los procesos, desde la adquisición hasta la entrega.

El machine learning ha demostrado tener un gran potencial en varios ámbitos, incluida la configuración de optimización del control de inventario. La optimización de inventario es fundamental para la gestión eficiente de los minoristas, independientemente de su tamaño. Implica manejar una gran cantidad de productos de manera regular con el objetivo de reducir costos operativos y aumentar las ventas. Una parte esencial de este proceso es el control de inventario, que implica decidir cuándo y cuánto pedir de un artículo en particular para mantener un equilibrio óptimo entre oferta y demanda. La gestión efectiva del inventario es crucial para las empresas, ya que impacta tanto en la disponibilidad del producto como en los costos operativos. Aunque reducir los tiempos de revisión y aumentar los pedidos puede mejorar la disponibilidad y reducir la pérdida de ventas, también puede aumentar los costos de inventario, incluidos los de mantenimiento y pedidos. Por ello, es esencial optimizar meticulosamente las políticas de inventario, ajustándolas según parámetros como el periodo de revisión, el tiempo de entrega y el nivel de servicio objetivo para cada artículo. El Machine Learning, al aprovechar algoritmos avanzados para analizar datos, identificar patrones y hacer predicciones precisas, revoluciona la gestión de inventario. Puede manejar relaciones no lineales, incorporar factores externos como el clima o tendencias de redes sociales, y ajustar los pronósticos en tiempo real según las condiciones del mercado. Esto mejora la precisión de la previsión de la demanda y optimiza los niveles de inventario en consecuencia. Con su capacidad para aprender y adaptarse a partir de datos, el Machine Learning es más dinámico y flexible que el software tradicional de gestión de inventario, especialmente en la previsión y optimización de la demanda. Puede manejar patrones de demanda complejos, identificar tendencias y ajustar pronósticos según múltiples variables y restricciones. Esto permite a las empresas determinar puntos de reabastecimiento óptimos, niveles de existencias de seguridad y estrategias de asignación de inventario, lo que resulta en decisiones de gestión de inventario más precisas, una mejor gestión de la cadena de suministro, y una reducción del riesgo de desabastecimiento o exceso de inventario.

Implementar Machine Learning en la gestión de inventario ofrece una serie de beneficios significativos: Por un lado, la Mejora de la Precisión de Pronósticos. Los algoritmos de Machine Learning permiten determinar niveles óptimos de inventario considerando factores como tiempo de entrega, estacionalidad y limitaciones de costos. Identifican el equilibrio adecuado entre costos de mantenimiento y desabastecimientos, optimizando los niveles de inventario a través del análisis de datos históricos, ciclos de producción y pronósticos de ventas. Por otro lado la Reducción de Costos y Pérdidas, es decir, analizan datos históricos de ventas, tendencias del mercado y factores externos para pronosticar con precisión la demanda de los clientes. Permite generar pronósticos de demanda más precisos, optimizando los niveles de inventario, reduciendo desabastecimientos y evitando excesos de inventario. Por último, también optimiza el ciclo de vida del producto. Los algoritmos de Machine Learning pueden generar probabilidades asociadas con diferentes niveles de demanda, permitiendo una gestión más eficiente de los productos. Por ejemplo, en una tienda de comestibles, pueden analizar fechas de vencimiento, patrones de demanda y datos históricos de ventas para optimizar los niveles de stock y minimizar el desperdicio debido a la caducidad del producto.

Estas son varias consideraciones y pasos clave para implementar Machine Learning en la gestión de inventario. La evaluación de necesidades, es decir, reflexionar sobre objetivos claros y específicos, como evitar el exceso de existencias, prevenir desabastecimientos o mejorar la previsión de la demanda. La selección de Modelos y Algoritmos, seleccionado modelos como regresión lineal, árboles de decisión o redes neuronales según la complejidad del problema y los datos disponibles. Entrenar los modelos con datos históricos, evaluando su rendimiento con métricas adaptadas a los objetivos de gestión de inventario, y ajustar y optimizar el modelo para alcanzar un rendimiento óptimo. Y por último, la integración con Sistemas Existentes, Colaborando estrechamente entre analistas de datos, profesionales de IT y expertos en gestión de suministros para integrar el modelo en el sistema de gestión de inventario. Alinear los resultados del modelo con los procesos existentes y garantizar una integración fluida en la toma de decisiones. Establecer un proceso de monitoreo y mantenimiento constante, dado que los modelos de Machine Learning son dinámicos y requieren actualizaciones periódicas.

Empresas líderes como Amazon, Walmart y Nike han implementado con éxito el Machine Learning en la gestión de inventario, obteniendo resultados significativos. En el caso de Amazon por ejemplo, utiliza el Machine Learning para predecir la demanda de productos considerando diversas variables como tendencias de búsqueda, datos históricos de ventas y condiciones climáticas. Esto le permite mantener niveles óptimos de inventario, evitando excesos o faltantes de stock. Para el caso de Walmart, emplea un sistema de gestión de inventario impulsado por IA para proporcionar a los clientes lo que necesitan, cuando lo necesitan y al costo esperado. Combina datos históricos con análisis predictivos para colocar estratégicamente los artículos en centros de distribución y tiendas, optimizando la experiencia de compra. Y por último, Nike, está construyendo una cadena de suministro digital a nivel mundial para atender directamente a los consumidores a escala[10].

## Objetivos

El objetivo de este proyecto es crear un cuadro de mando donde se pueda ver la información necesaria en cada momento para mejorar las prácticas de gestión de inventarios de la empresa. Es decir, aprovechar el análisis exhaustivo de datos para optimizar el control del inventario y extraer información valiosa del funcionamiento de la empresa, en particular de las ventas y las compras. Las tareas del proyecto se dividen en dos objetivos principales:

* Realizar un preprocesamiento efectivo de los datos de inventario utilizando **Python** para garantizar la calidad y la integridad de los datos antes de su análisis.
* Implementar scripts de Python que limpien, formateen y preparen los datos de inventario para su análisis.
* Utilizar bibliotecas de Python como Pandas y NumPy para realizar el preprocesamiento de datos de manera eficiente.
* Garantizar que los datos estén libres de errores y sean coherentes para un análisis preciso.
* Crear un informe interactivo en Power BI que aborde los problemas de gestión de inventario identificados y proporcione una visión clara del estado del inventario en todo momento.
* Desarrollar un informe en Power BI que incluya visualizaciones dinámicas de datos de inventario, métricas clave y recomendaciones para mejorar la gestión de inventario.
* Utilizar las capacidades de Power BI para crear visualizaciones efectivas y personalizadas que aborden los problemas específicos de gestión de inventario.
* Implementar un control de versiones utilizando GitHub para gestionar y rastrear cambios en el código y los recursos del proyecto de análisis de inventario.
* Crear un repositorio en GitHub y realizar commits periódicos para registrar los cambios en el código y los archivos relacionados con el proyecto.

## Descripción de la metodología empleada

En términos de herramientas de software, este proyecto sigue una metodología centrada en el uso de Python para el procesamiento de datos. Se emplea un entorno Jupyter Notebook (.ipynb) para llevar a cabo todas las fases del tratamiento de datos, incluyendo carga, preprocesamiento y modelado. Además, para la creación del cuadro de mando, se utiliza PowerBI. En cuanto a la documentación y presentación del proyecto, la redacción se realiza en Microsoft Word y la presentación en Microsoft PowerPoint. Estas herramientas han sido seleccionadas debido a su idoneidad para un proyecto nuevo que se desarrolla completamente desde cero. En relación con las diferentes etapas que abarca este proyecto, se pueden distinguir las siguientes:

* Carga de datos: En la fase inicial del proyecto, los datos se recopilan inicialmente en archivos en formato .csv o Excel. Estos archivos se transforman en dataframes utilizando Python dentro del entorno Jupyter Notebook.
* Preprocesamiento de los datos: Durante esta etapa inicial, se lleva a cabo el procesamiento de los datos con el objetivo de asegurar su calidad y prepararlos adecuadamente para las tareas de análisis. Esto incluye la limpieza de los datos, el tratamiento de valores nulos y la normalización de datos cuando es necesario
* Análisis de los datos: En la fase de análisis de datos, se ha realizado un esfuerzo por comprender exhaustivamente los datos disponibles. Se han generado nuevas tablas que añaden valor al informe y se han creado columnas adicionales a partir de los datos existentes para enriquecer la información disponible
* Creación del informe: Las bases de datos almacenadas en Excel han sido integradas en Power BI, donde se ha procedido a crear un informe dinámico y claro. Este proceso asegura la correcta visualización de los datos de manera accesible y comprensible para los usuarios.
* Control de versiones: Durante todo el proyecto se ha hecho uso se git para mantener una correcta gestión de las versiones.
* Conclusiones: una vez completado todo el proceso se extraen las conclusiones definitivas.
* Redacción de la memoria: Este proceso se extiende a lo largo de todo el proyecto de fin de máster, comenzando con la selección del proyecto y abarcando hasta la conclusión del análisis de datos y la obtención de resultados.

## Planificación

(Insertar Gantt)

# DESARROLLO

Para llevar a cabo el desarrollo del proyecto se han dado una serie de pasos interrelacionados que van desde la entrada de datos hasta la visualización de los resultados. En el primer paso se han ingestado los datos, lo que implica la recopilación de información relevante. A continuación, se ha seguido con la preparación de datos, lo que implica la limpieza y consolidación de datos brutos para transformarlos en una forma adecuada para el análisis. Se verifica continuamente para garantizar la calidad y relevancia de los datos. Luego, en la exploración de datos se ha estudiado el conjunto de datos y se ha enriquecido con información adicional para proporcionar una perspectiva más completa y revelar insights. También es esencial la combinación de datos, software y procesos comerciales junto con la intuición humana que permite obtener insights accionables que respaldan la toma de decisiones empresariales, que es de donde viene el termino Business Intelligence. Y por último, se ha generado el informe donde se organizan y presentan los resultados del análisis de manera efectiva para compartir los conocimiento adquiridos de manera comprensible y útil.

## Análisis de los datos

Para llevar a cabo el proyecto se ha cogido la información de la plataforma llamada Kaggle que proporciona un entorno colaborativo para el aprendizaje automático y la ciencia de datos. Esta página ofrece una variedad de recursos y herramientas en los que entre otros se encuentran bases de datos públicos[11].

### Entrada de datos:

Para llevar a cabo la lectura de datos y el procesamiento, se ha optado por emplear el entorno de desarrollo integrado Visual Studio Code, aprovechando su funcionalidad y facilidad de uso en la programación en Python, un lenguaje ampliamente utilizado en el ámbito del análisis de datos y la ciencia de datos. En la fase inicial del proceso, se ha procedido a importar las bibliotecas y paquetes necesarios para la manipulación, análisis y visualización de datos. Entre estas herramientas esenciales, se incluyen Pandas, una biblioteca de manipulación y análisis de datos que proporciona estructuras de datos flexibles y eficientes, así como funciones para la lectura y escritura de datos en diversos formatos; NumPy, una biblioteca fundamental para la computación numérica en Python, que ofrece soporte para matrices y funciones matemáticas de alto nivel; Matplotlib, una biblioteca ampliamente utilizada para la visualización de datos en dos dimensiones, que permite la creación de gráficos estáticos, interactivos y animados; y Seaborn, una biblioteca basada en Matplotlib que proporciona una interfaz de alto nivel para la creación de gráficos estadísticos atractivos y informativos.La elección de estas herramientas se sustenta en su capacidad para facilitar las tareas de lectura, procesamiento y análisis de datos, así como en su versatilidad y potencia para generar visualizaciones claras y efectivas que permitan comprender y comunicar adecuadamente los resultados obtenidos durante el análisis de datos. Dicho esto, se han cargado los 6archivos Excel con los que se dispone para llevar adelante el trabajo. Estos corresponden a operaciones comerciales del ejercicio cerrado en 2016. El conjunto de datos incluye las siguientes bases de datos: Inventario inicial del 2016, inventario final del 2016, facturas de compra correspondientes a 2016, precios de compra, datos de compras y datos de ventas. Una vez cargados los datos se han impreso las primeras 5 lineas de todos los datos para obtener una visión general de los datos.

### Preparación de datos

Una vez leídos las bases de datos, se ha comenzado con el análisis exploratorio de ellos, obteniendo una tabla donde se han recogido detalles clave sobre los datos. Entre ellos, se han recogido los nombres de las columnas, el número de filas, el tipo de los datos, el número de valores únicos de cada columna, los valores no informados y el porcentaje de valores no informados frente al total. Así con la foto clave de cada tabla se han sacado varias conclusiones para después proceder a la limpieza y preprocesamiento de los datos.

Se ha visto que la columna *InventoryId* tiene los mismos registros únicos que los totales, lo que sugiere que podrían servir como claves primaria y foránea para unir las tablas. También se ha visto que varias tablas contienen valores no informados que es necesario identificar y tratar adecuadamente. Además, las columnas de tamaño muestran incoherencias en todas las tablas debido a la variación de unidades de medida. Por ello, es necesario convertirlas a una unidad única y coherente. Además de la columna de volumen, en alguna de las tablas también existe la columna “tamaño”, la cual contiene el mismo dato, por lo que se puede eliminar una de las dos columnas. Hablando del formato de las fechas se ha observado que utilizan diferentes formatos en cada una de las tablas por lo que es conveniente unificar y ponerlos de la misma manera. Y por último, en la columna *VendorName* se pueden ver espacios o signos tras la palabra que se pueden eliminar.

### Limpieza y procesamiento

En el apartado anterior se han mencionado cuales son los puntos que se deben abordar para llevar a cabo la limpieza y el procesamiento de los datos, por lo que a continuación se han producido dichos cambios.

#### Tratamiento de Missings

La integridad y la calidad de los datos son aspectos fundamentales en cualquier análisis de datos o proyecto de investigación. Los valores no informados, comúnmente conocidos como "missing values" en inglés, representan una preocupación significativa en el contexto del análisis de datos, ya que pueden distorsionar los resultados y conducir a interpretaciones erróneas si no se manejan adecuadamente. Por lo tanto, es crucial detectar y abordar de manera efectiva estos valores faltantes. Los valores no informados pueden surgir debido a una variedad de razones, que van desde errores en la entrada de datos hasta fallos en la recopilación o transferencia de información. Independientemente de su origen, la presencia de valores no informados puede comprometer la validez y la fiabilidad de cualquier análisis realizado sobre los datos afectados. Durante el análisis del conjunto de datos que comprende varias tablas, se han detectado  valores no informados en cuatro de las tablas analizadas. A continuación, se describen las acciones tomadas para abordar estos valores faltantes:

* Tabla “end\_inventory” : En la columna “City se ha identificado la ausencia de valores. Tras un análisis detallado, se ha observado que todos los nombres de ciudades están presentes excepto para una ubicación de tienda específica. Dado este hallazgo, se ha decidido asignar el nombre "TYWARDREATH" a esta ciudad, basándose en el número de tienda correspondiente.
* Tabla “purchases”: Se han encontrado solo tres valores faltantes en la columna "size", lo que representa un porcentaje mínimo del total de registros. Por lo tanto, se ha optado por por eliminar estas filas para preservar la integridad de los datos restantes.
* Tabla “sales”: En la columna "Approval" se muestra un alto porcentaje de valores faltantes, lo que representa el 93% del total de registros. Dada la magnitud de esta ausencia de información, se ha tomado la decisión de eliminar completamente esta columna para evitar cualquier sesgo o distorsión en el análisis posterior.
* Tabla purchase\_prices”: Se han identificado valores faltantes en las columnas "Description", "Size" y "Volume". Dado que estos valores no informados representan registros individuales, se ha decidido eliminar las filas correspondientes para mantener la coherencia y la integridad de los datos restantes.

#### Tratamiento de duplicados

Mediante el método duplicated en Python, se ha realizado una verificación para determinar la presencia de duplicados dentro del conjunto de datos representado. Se ha guardado la información en una variable llamada duplicados, la cual captura el resultado booleano, proporcionando una indicación clara sobre la presencia o ausencia de duplicados en los datos. Este enfoque es esencial para garantizar la integridad y la calidad de los datos, ya que la presencia de duplicados puede distorsionar los resultados del análisis y conducir a conclusiones erróneas si no se manejan adecuadamente.

#### Irregularidades en los datos

Se han observado irregularidades significativas en las entradas de datos, especialmente en la columna "Size", que exhibe variaciones en todas las tablas de datos, con la excepción del conjunto de datos "purchase\_price". Estas discrepancias se refieren a la representación de información de volumen, expresada en diferentes unidades y formatos, como litros, mililitros, onzas, paquetes (pk) y combinaciones de estas unidades. Este panorama heterogéneo implica la necesidad de estandarizar estas medidas en una unidad de volumen única y coherente.

Para abordar esta cuestión, se ha llevado a cabo un análisis exhaustivo de los recuentos únicos en la columna "Size". Aquellos elementos que exhiben un solo recuento y no presentan un patrón discernible han sido asignados a una tasa equivalente estándar. Por otro lado, aquellos elementos con múltiples recuentos y un patrón discernible han sido transformados de acuerdo con estos patrones identificados.

Este enfoque metodológico busca garantizar la consistencia y la comparabilidad de los datos de volumen a lo largo de las diferentes tablas, lo que es esencial para facilitar un análisis coherente y significativo en el contexto de la investigación o análisis de datos en cuestión. La estandarización de las unidades de volumen contribuye a mitigar posibles sesgos o distorsiones en los resultados del análisis, permitiendo así una interpretación precisa y fiable de los datos.

Además, para garantizar la coherencia y la fiabilidad en el manejo de datos temporales, se ha llevado a cabo una normalización del formato de fecha y hora en un DataFrame específico. Es importante destacar que las columnas de fecha en los conjuntos de datos de "Inventario inicial" e "Inventario final" ya se encuentran en un formato adecuado, por lo que no requieren ninguna manipulación adicional. En consecuencia, nos enfocaremos en formatear las columnas de fecha en los demás conjuntos de datos.

Este proceso de normalización del formato de fecha y hora tiene como objetivo principal garantizar la cohesión y la consistencia en el tratamiento de datos temporales, lo que facilita su posterior análisis y visualización. Al estandarizar el formato de fecha y hora en todas las columnas pertinentes del DataFrame, se promueve la precisión y la interpretabilidad de los datos, lo que es esencial para obtener conclusiones sólidas y confiables en cualquier análisis posterior.

Por último, se ha identificado una discrepancia en los nombres de las ciudades presentes en las tablas de "Inventario inicial" y "Inventario final", donde los nombres no corresponden a ubicaciones geográficas reales. Para abordar esta irregularidad, se ha realizado un proceso de corrección mediante el reemplazo de los nombres de las ciudades no reales por aquellos que sí lo son. Para llevar a cabo esta corrección, se han extraído todos los nombres de ciudades presentes en las tablas y mediante la función replace se han reemplazado con una lista de ciudades geográficamente válidas. Para complementar la información, se ha añadido una nueva columna llamada estado que indica el estado de cada ciudad. Esta información adicional se ha considerado apropiada para visualizarlo en el informe mediante un mapa geográfico. Además, se ha observado que el identificador de inventario (InventoryId) estaba influenciado por el nombre de la ciudad. Por lo tanto, se ha procedido a recrear este identificador combinando el número de tienda, el nombre de la ciudad y el número de marca, separados por un guion. Este proceso de recreación del identificador se ha aplicado en las tablas pertinentes donde se encontraba esta relación entre la ciudad y el InventoryId. Este enfoque de corrección y recreación de registros asegura la coherencia y la precisión de los datos, lo que facilita su interpretación y análisis posteriores. Además, garantiza que los identificadores de inventario reflejen de manera precisa la relación entre la tienda, la ciudad y la bodega, proporcionando así una representación más fiel de la realidad en el conjunto de datos. Para terminar, se han impreso otra vez todas las tablas, y se ha asegurado que se ha completado la limpieza y procesamiento de los datos, para así empezar con la siguiente fase.

Una vez teniendo los datos limpios y procesados, se han creado nuevas tablas con la intención de posibilitar y facilitar la creación del modelo. Estas tablas serán las que serán ingestadas en PowerBI para realizar el modelo. A la hora de crear las tablas necesarias, se ha tenido en mente el modelo semántico que se comenta más adelante y la información que se quiero mostrar. Una vez teniendo eso en mente, se han cogido las columnas más relevantes de las tablas que se han leido y se han creado tablas nuevas que han sido exportadas a excel para luego poder ingestarlas en PowerBI.

Vista la importancia de crear el modelo semántico correcto, se ha profundizado en ello antes de mostrar las tablas.

## Modelo semántico en powerbi:

La creación de un modelo semántico excelente es una de las tareas más importantes ya que al hacer este trabajo de forma correcta, ayudará a los usuarios a comprender mejor los datos, lo que facilitará la creación de informes de Power BI útiles para ellos y para el creador. Un buen modelo semántico ofrece las siguientes ventajas: una exploración de datos más rápida, agregaciones más fáciles de crear, informes más precisos, dedicar menos tiempo al crear el informe y mejor seguimiento a futuro[12].

A la hora de crear el modelo, se ha optado por el modelo estrella. Aunque no es la única manera de simplificarlos, es un método popular para simplificar los datos. En un esquema de estrella, cada tabla del modelo semántico se define como una tabla de hechos o de dimensiones, como se muestra en la siguiente Figura 1.

Figura 1: Modelo estrella

Diagrama, Forma

Descripción generada automáticamente con confianza media

Fuente: [12]

El esquema en estrella se fundamenta en una tabla de hechos que contiene uno o más datos que en lo posible deben ser aditivos o mensurables y una o más tablas de dimensiones. La llave primaria de la tabla de hechos es una concatenación de las llaves primarias de cada una de las dimensiones. Los hechos, con frecuencia datos agregados, pueden pensarse como medidas tomadas de la intersección de todas las dimensiones. Los atributos en las dimensiones son, por lo general, textuales y discretos, con el propósito de establecer restricciones específicas en las consultas. La construcción de una base de datos bajo este esquema requiere de una rígida definición de hechos y dimensiones (muchas veces sin atender las reglas de normalización) que anticipen la consulta de datos bajo patrones preestablecidos[13].

Las tablas de hechos contienen valores de datos de eventos o de observación y pueden contener varios valores repetidos. Por ejemplo, un producto puede aparecer varias veces en varias filas para diferentes registros. Con las tablas de hechos, es habitual ver columnas rellenadas con números y fechas. Las tablas de dimensiones contienen los detalles sobre los datos de las tablas de hechos: productos, ubicaciones, clientes y proveedores. Estas tablas están conectadas a la tabla de hechos a través de columnas de clave. Las tablas de dimensiones se usan para filtrar y agrupar los datos de las tablas de hechos. Volviendo al modelo estrella, que es como se ha definido el modelo, a continuación, se explica cómo se han creado las relaciones entre las tablas en powerBI[14].

Al tener varias tablas, las relaciones entre ellas son necesarias para calcular los resultados de forma precisa y mostrar la información correcta en los informes. En caso de que se hayan nombrado correctamente las columnas, es decir, poniendo el mismo nombre a aquellas que se quieren relacionar, la característica de detección automática es capaz de crear las relaciones automáticamente. Sin embargo, es necesario revisarlas todas y realizar cambios en los casos que sea necesario. Además, al crear o editar una relación, se puede configurar más de una opción. La opción cardinalidad puede tener una de las siguientes configuraciones:

* Varios a uno (\*:1): Esta es la relación predeterminada más común. Indica que una columna de una tabla puede contener múltiples instancias de un valor, mientras que la tabla relacionada, frecuentemente denominada tabla de búsqueda, tiene solo una instancia de ese valor.
* Uno a uno (1:1): En este tipo de relación, la columna de una tabla contiene una única instancia de un valor específico, y la tabla relacionada también contiene una única instancia de ese mismo valor.
* Uno a varios (1:\*): En una relación de uno a varios, la columna de una tabla contiene una sola instancia de un valor, y la tabla relacionada puede contener múltiples instancias de ese mismo valor.
* Varios a varios (:): Los modelos compuestos permiten establecer relaciones de varios a varios entre tablas, eliminando la necesidad de valores únicos en las tablas. Esto también facilita evitar soluciones alternativas previas, como la creación de nuevas tablas únicamente para establecer relaciones. Para obtener más información, consulte "Relaciones con una cardinalidad de varios a varios".

La opción Dirección de filtro cruzado también es muy importante a la hora de crear las relaciones que puede tener unas de las siguientes opciones

* Ambos: Esta configuración indica que, para efectos de filtrado, ambas tablas se tratan como si fueran una sola tabla. Es ideal para una única tabla rodeada de muchas tablas de búsqueda, como una tabla central de datos de ventas con una tabla de búsqueda de departamentos. Esta configuración es comúnmente conocida como esquema de estrella (una tabla central con varias tablas de búsqueda). No obstante, si existen dos o más tablas que también tienen tablas de búsqueda compartidas, no es recomendable utilizar la opción Ambos.
* Único: Esta es la dirección predeterminada más común, donde las opciones de filtrado en las tablas conectadas afectan solo a la tabla en la que se agregan los valores[14].

Por último, recalcar la importancia de las claves primarias y foráneas en el contexto de Power BI para definir y gestionar las relaciones entre las tablas de un modelo de datos. Una clave primaria es una columna o un conjunto de columnas en una tabla cuyos valores identifican de forma exclusiva una fila de la tabla. Una base de datos relacional está diseñada para imponer la exclusividad de las claves primarias permitiendo que haya sólo una fila con un valor de clave primaria específico en una tabla. En este caso las claves primarias se han nombrado indicando un id por delante. En cuanto a la clave foránea, es una columna o un conjunto de columnas en una tabla cuyos valores corresponden a los valores de la clave primaria de otra tabla. Para poder añadir una fila con un valor de clave foránea específico, debe existir una fila en la tabla relacionada con el mismo valor de clave primaria[15].

Tras entender la importancia del modelo semántico y la creación y administración de las relaciones, a continuación, se ha mostrado una tabla de resumen para tener una visión general de las tablas ingestadas en powerbi. Aquellas tablas que contienen una D por delante significa que son tablas de dimensiones, y las que contienen una H, tablas de hechos.

Tabla 1: Descripción de las tablas del modelo

|  |  |  |
| --- | --- | --- |
| **Tabla** | **Columna** | **Descripción** |
| D\_PRODUCTO | IdProducto | Clave primaria. Es un denominador único de la tabla que se utiliza para relacionar con la tabla de hechos. |
| Producto | Nombre del producto |
| Volumen | El volumen en litros de cada producto |
| Precio unidad | El precio de la unidad de cada producto |
| Categoría | Se han dividido 5 categorías (Muy bajo, Bajo, Medio, Alto,Muy Alto) dependiendo del precio unidad de cada producto |
| D\_CLIENTES | IdCliente | Clave primaria. Es un denominador único de la tabla que se utiliza para relacionar con la tabla de hechos |
| Cliente | Nombre del cliente |
| D\_PROVEEDORES | IdProveedor | Clave primaria. Es un denominador único de la tabla que se utiliza para relacionar con la tabla de hechos |
| Proveedor | Nombre del proveedor |
| D\_TIENDAS | IdTienda | Clave primaria. Es un denominador único de la tabla, que se utiliza para relacionar con la tabla de hechos |
| Ciudad | Nombre de la ciudad en la que se encuentra la tienda, ya que hay una única tienda por ciudad |
| Estado | El nombre del estado de la ciudad donde se encuentra la tienda |
| D\_CALENDARIO \* | Fecha | Clave primaria. Fecha completa construida mediante la función CALENDAR() |
| Año | Año de la fecha |
| MesNum | Número del mes de la fecha |
| SemanaNum | Número de la semana de la fecha indicada |
| DiaSemana | Día de la semana correspondiente |
| Mes | Mes correspondiente |
| Día | Número del día correspondiente del mes |
| H\_VENTAS | IdTienda | Clave foránea. Código de la tienda que se relaciona con la tabla M\_TIENDAS |
| IdProducto | Clave foránea. Código del producto que se relaciona con la tabla M\_PRODUCTOS |
| IdCliente | Clave foránea. Código del cliente que se relaciona con la tabla M\_CLIENTE |
| Fecha venta | Fecha de la venta de producto |
| Cantidad ventas | Cantidad de ventas realizadas |
| Precio unidad | Precio unitario del producto vendido |
| Precio venta | Precio de venta considerando la cantidad vendida |
| Impuesto | Precio del impuesto |
| Precio total | Columna calculada. Precio total considerando la cantidad y el impuesto |
| H\_COMPRAS\_FILTRADA | IdTienda | Clave foránea. Código de la tienda que se relaciona con la tabla M\_TIENDAS |
| IdProducto | Clave foránea. Código del producto que se relaciona con la tabla M\_PRODUCTOS |
| IdProveedor | Clave foránea. Código del proveedor que se relaciona con la tabla M\_PROVEEDOR |
| Fecha orden de compra | Fecha de cuando se realizó la compra |
| Fecha llegada | Clave foránea. Fecha de la llegada del producto a planta, que se relaciona con M\_CALENDARIO |
| Fecha factura | Fecha de la factura |
| Fecha de pago | Fecha de pago |
| Precio compra unidad | El precio unitario de compra |
| Cantidad | La cantidad de compra |
| Precio compra total | El precio total de la compra, considerando la cantidad |
| Plazo de entrega | Columna calculada, indica el plazo desde que se realizó la orden de compra hasta que llega el producto. |
| H\_INVENTARIO | IdTienda | Clave foránea. Código de la tienda que se relaciona con la tabla M\_TIENDAS |
| IdProducto | Clave foránea. Código del producto que se relaciona con la tabla M\_PRODUCTOS |
| Stock inicial | Cantidad de stock el día 01/01/2016 |
| Fecha stock inicial | Fecha cuando se contabilizó el stock |
| Stock final | Cantidad de stock el día 31/12/2016 |
| Fecha stock final | Fecha cuando se contabilizó el stock |
| H\_INVENTARIO\_INICIAL | IdTienda | Clave foránea. Código de la tienda que se relaciona con la tabla M\_TIENDAS |
| IdProducto | Clave foránea. Código del producto que se relaciona con la tabla M\_PRODUCTOS |
| Stock inicial | Cantidad de stock el día 01/01/2016 |
| Fecha | Fecha cuando se contabilizó el stock |
| H\_ANÁLISIS\_ABC | IdProducto | Clave foránea. Código del producto que se relaciona con la tabla M\_PRODUCTOS |
| Coste unidad | Coste del producto por unidad |
| Venta unidad | Precio de la venta por unidad |
| Cantidad ventas | Número de cantidad de ventas |
| Precio venta | Precio de ventas considerando la cantidad |
| Ratio | Relación entre las ventas de cada artículo y las ventas totales |
| Categoría | Categoría definida (A,B,C) |
| Beneficio unidad | El beneficio por unidad, es decir, la resta entre el precio de venta y el precio de compra |

La tabla D\_CALENDARIO donde se guardan las fechas para después relacionarla con las tablas de hecho, se ha creado utilizando DAX. Se ha usado la función de expresión de análisis de datos (DAX) CALENDAR() para crear la tabla de fechas común. CALENDAR() devuelve un intervalo de fechas contiguo en función de fechas de inicio y de finalización que se especifican como argumentos en la función. La fecha de inicio se elige como la más temprana del modelo semántico y la fecha de finalización es la última, además de los datos que se han rellenado para el mes fiscal. Una vez creada la tabla se cuenta con una columna de fechas que se puede usar. Pero esta columna es ligeramente dispersa. También quiere ver columnas solo del año, el número del mes, la semana del año y el día de la semana. Por ello, se ha realizado esta tarea creando nuevas columnas y escribiendo la ecuación DAX, que recuperará el año de la tabla de fechas[16].

## Informe PowerBI

<https://learn.microsoft.com/es-es/power-bi/create-reports/service-dashboards-design-tips>

Crear el modelo, Crear medidas DAX, Fondo y estructura

## Aplicación de Modelos para la predicción de la demanda

Después de revisar varios modelos de machine learning en el marco teórico, se ha procedido a aplicar un ejemplo de modelo simple para la licorería siguiendo un enfoque práctico. Para ilustrar este proceso, se ha seleccionado un modelo de los previamente mencionados y se ha implementado para predecir la demanda utilizando datos históricos disponibles.

A la hora de crear el modelo de predicción, primero de todo, se ha realizado la consolidación de datos, donde se han combinado los datos de ventas y compras en un único dataset para facilitar el análisis. En ambos casos, se han cogido los datos desde el 01-01-2016 hasta el 29-02-2016. Luego se ha realizado un Análisis Exploratorio de Datos (EDA) para entender mejor las características de los datos y detectar p patrones, tendencias y posibles anomalías. En este caso se ha visualizado la serie temporal de ventas y compras, como se ve en la siguiente Figura 2.

Figura 2:Serie temporal de ventas y compras

Gráfico, Histograma

Descripción generada automáticamente

A continuación, se han generado nuevas características que puedan ser útiles para el modelo de predicción, como tasas de ventas diarias, ventas y compras promedio por semana/mes etc. Luego se ha seleccionado el modelo de predicción, donde al principio se ha realizado un modelo de regresión, pero luego se ha aplicado un modelo de series temporales viendo que puede tener patrones estacionales. Seguido se ha entrenado el modelo utilizando los datos disponibles y se ha validado su desempeño. En esta fase, primero se han dividido los datos en conjuntos de entrenamiento y validación. Y luego se ha entrenado el modelo seleccionado con los datos de entrenamientos. Por último, se ha evaluado el modelo utilizando métricas de evaluación las cuales han sido MAE (Mean Absolute Error) y RMSE (Root Mean Square Error). La métrica de evaluación MAE (Mean Absolute Error o Error Absoluto Medio) es una medida de la precisión de un modelo de predicción. Representa la media de las diferencias absolutas entre las predicciones del modelo y los valores reales observados. Un valor de MAE más bajo indica un modelo más preciso. En este caso, tras realizar el modelo de regresión lineal, el valor MAE: 224483.14. Esto significa que, en promedio, las predicciones del modelo están desviadas de los valores reales por aproximadamente 224483.14 unidades de la cantidad. El error cuadrático medio de la raíz (RMSE, por sus siglas en inglés, Root Mean Squared Error) es una medida comúnmente utilizada para evaluar la precisión de un modelo de regresión o de predicción. Se calcula como la raíz cuadrada de la media de los errores al cuadrado entre los valores predichos por el modelo y los valores observados reales. Este mide la diferencia promedio entre los valores predichos y los valores reales en la misma escala que los datos originales. Cuanto más bajo sea el valor de RMSE, mejor será la capacidad predictiva del modelo. En este caso el valor obtenido es 224578.26. Al tener una escala de datos muy grande, es decir, los datos de ventas y compras son de gran magnitud, puede llegar a ser aceptable pero aun así se ha intentado mejorar realizando otro modelo de predicción más avanzado como ARIMA, SARIMA, o modelos de aprendizaje automático como Random Forest, Gradient Boosting, etc.

Después de encontrar resultados subóptimos con un enfoque de regresión lineal para predecir la demanda, se ha decidido explorar la eficacia del modelo de Random Forest. Este modelo más avanzado ofrece la capacidad de manejar relaciones no lineales y capturar la complejidad de los datos de manera más efectiva. En este caso, primero de todo se han importado las bibliotecas necesarias: “RandomForestRegressor” para el modelo de regresión forestal aleatoria, “GridSearchCV” para la búsqueda de hiperparámetros, y “train\_test\_split” para dividir los datos en conjuntos de entrenamiento y prueba. A continuación, se han definido las características (x) y la variable objetivo (y) para el modelo y se ha dividido el conjunto de datos en conjuntos de entrenamiento y prueba con una proporción del 20% de los datos reservados para pruebas y un estado aleatorio fijado en 42. Para obtener el resultado más optimo, se ha definido un diccionario de hiperparámetros para probar diferentes combinaciones de valores. Se ha realizado una búsqueda de hiperparámetros utilizando “GridSearchCV”, con validación cruzada de 3 pliegues y métrica de evaluación configurada en el error absoluto medio negativo. Después, se ha ajustado el modelo con los mejores hiperparámetros encontrados durante la búsqueda de hiperparámetros. Una vez obtenidos los mejores hiperparámetros, se han realizado las predicciones en el conjunto de prueba utilizando el mejor modelo. Y por último se han calculado el error absoluto medio (MAE) y el error cuadrático medio (RMSE) entre las predicciones y los valores reales en el conjunto de prueba. Se imprimen los resultados de MAE y RMSE. Estos valores indican la diferencia promedio absoluta y la raíz del error cuadrático medio entre las predicciones del modelo y los valores reales en el conjunto de prueba, respectivamente. Los resultados del modelo Random Forest muestran una mejora significativa en comparación con el enfoque de regresión lineal anterior. El error absoluto medio (MAE) y el error cuadrático medio de la raíz (RMSE) indican que las predicciones del modelo tienen un desempeño prometedor, con un MAE de aproximadamente 60981 unidades y un RMSE de alrededor de 69531 unidades. Estas métricas son esenciales para evaluar la precisión del modelo, y valores más bajos indican una mejor capacidad de predicción. Además, los coeficientes de determinación, tanto en el conjunto de entrenamiento (0.889) como en el conjunto de prueba (0.914), son altos. Esto sugiere que una gran parte de la variabilidad en la variable dependiente puede ser explicada por las características incluidas en el modelo. Un coeficiente de determinación cercano a 1 indica un buen ajuste del modelo a los datos, lo que confirma la capacidad predictiva del modelo Random Forest. En resumen, estos resultados indican que el modelo Random Forest ha logrado capturar de manera efectiva la relación entre las variables predictoras y la variable objetivo, superando notablemente el desempeño del modelo de regresión lineal anterior. Esto respalda la elección del modelo Random Forest como un enfoque más adecuado para la tarea de predicción en este contexto específico.

En este estudio, además de explorar el rendimiento del modelo Random Forest, también se ha considerado la aplicación de un enfoque alternativo utilizando un modelo ARIMA. Sin embargo, antes de proceder con el modelo ARIMA, se ha llevado a cabo un análisis para determinar si la serie temporal subyacente es estacionaria o no. Esto implica que la media y la varianza de la serie no cambian con el tiempo. Este paso es crucial, ya que los modelos ARIMA requieren que la serie sea estacionaria para producir predicciones precisas. En este sentido, se ha investigado la estacionariedad de la serie temporal a través de la Prueba de Dickey-Fuller aumentada (ADF test). Este análisis proporciona información fundamental sobre la naturaleza de los datos y orienta la selección del modelo más apropiado para su modelado.

Figura 3: Ventas diarias

Gráfico, Histograma

Descripción generada automáticamente

La prueba de Dickey-Fuller aumentada (ADF) se ha utilizado para determinar si la serie temporal es estacionaria o no. La estadística ADF obtenida es -0.6876. Para decidir sobre la estacionariedad, se ha comparado esta estadística con los valores críticos correspondientes a los niveles de significancia del 1%, 5% y 10%. El valor p de la prueba es 0.8499. Un valor p alto (mayor que el nivel de significancia comúnmente usado como el 5%) indica que no se puede rechazar la hipótesis nula, lo que sugiere que la serie no es estacionaria. En resumen, dado que la estadística ADF es mayor que los valores críticos en todos los niveles de significancia y el valor p es significativamente alto, se puede concluir que la serie temporal no es estacionaria.

Cuando la serie temporal no es estacionaria, es necesario realizar algunas transformaciones para convertirla en estacionaria antes de poder aplicar modelos de series temporales como ARIMA de manera efectiva. Entre las estrategias más comunes existentes, se ha aplicado la de remover la estacionalidad. Esto se ha realizado restando la media móvil estacional o tomando la diferencia entre una observación y su valor en el mismo punto en el ciclo estacional anterior.

En el código, se ha calculado la media móvil estacional con una ventana de 7 días. Luego, se ha restado la media móvil estacional de la serie temporal original para eliminar la estacionalidad y obtener una nueva serie temporal. Finalmente, se ha visualizado tanto la serie temporal original como la serie sin estacionalidad para comparar cómo se ven en la Figura 4.

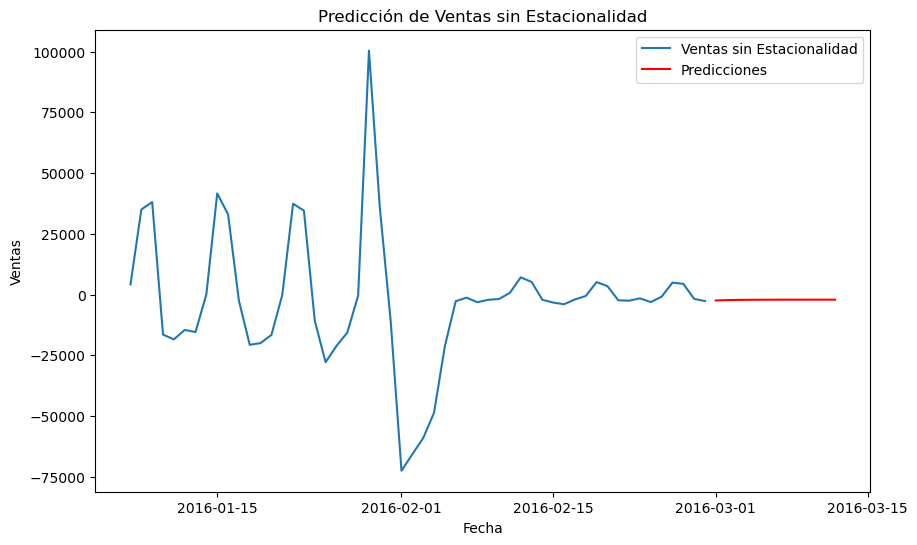
Figura 4: Ventas diarias con y sin estacionalidad

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Después de eliminar la estacionalidad, se ha procedido a ajustar un modelo ARIMA a la serie temporal para realizar predicciones. En el código, se ha ajustado un modelo ARIMA a la serie temporal sin estacionalidad utilizando la clase ARIMA de statsmodels. Después de entrenar el modelo, se han hecho las predicciones para los próximos 12 pasos adelante y se han visualizado en la Figura 5 dichas predicciones junto con la serie temporal original .

Figura 5: Predicción de ventas sin estacionalidad



# RESULTADOS

Xxxx

# CONCLUSIONES Y DISCUSIÓN

xxxxxxxx

**BIBLIOGRAFÍA**

[1] Jordi de Mas Jaumot, “Gestión del inventario en una empresa del sector farmacéutico mediante algoritmos de Machine Learning,” 2021. [Online]. Available: https://openaccess.uoc.edu/bitstream/10609/129826/8/jde\_masTFM0121memoria.pdf

[2] I. de I. Conocimiento, “Machine Learning & Deep Learning,” 2024, [Online]. Available: https://www.iic.uam.es/inteligencia-artificial/machine-learning-deep-learning/

[3] T. Šustrová, “A Suitable Artificial Intelligence Model for Inventory Level Optimization Tereza Šustrová: A Suitable Artificial Intelligence Model for Inventory Level Optimization,” vol. 25, no. 1, pp. 48–55, 2016, [Online]. Available: http://dx.doi.org/10.13164/trends.2016.25.48

[4] S. Zhang, X. Qin, S. Hu, Q. Zhang, B. Dong, and J. Zhao, “Importance Degree Evaluation of Spare Parts Based on Clustering Algorithm and Back-Propagation Neural Network,” *Math. Probl. Eng.*, vol. 2020, 2020, doi: 10.1155/2020/6161825.

[5] K. B. Praveen, P. Kumar, J. Prateek, G. Pragathi, and P. M. J, “Inventory Management System Using Machine Learning,” *Int. J. Innov. Eng. Manag. Res.*, vol. 9, no. 06, pp. 769–785, 2022, doi: 10.48047/ijiemr/v11/i06/51.

[6] Scikit learn, “Gradient Boosting Regressor,” 2024. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html

[7] “Decision tree in Machine Learning,” 2024, [Online]. Available: https://www.geeksforgeeks.org/decision-tree-introduction-example/

[8] Al Maverick, “Light Gradient Boosting Machine,” 2023. https://samanemami.medium.com/light-gradient-boosting-machine-b4f1b9e3f7d1

[9] Instituto de ingenieria del conocimiento, “Machine Learning y optimización para la gestión del stock,” 2024, [Online]. Available: https://www.iic.uam.es/noticias/machine-learning-y-optimizacion-para-gestion-stock/

[10] EAE Business School, “Descubre la Optimización de Inventario con Machine Learning,” 2024. https://www.eaemadrid.com/es/blog/machine-learning

[11] Kaggle, “Inventory Analysis case study,” 2023. https://www.kaggle.com/datasets/bhanupratapbiswas/inventory-analysis-case-study

[12] Microsoft, “Diseño de un modelo semántico en PowerBI,” 2023. https://learn.microsoft.com/es-es/training/modules/design-model-power-bi/1-introduction

[13] J. Quiroz, “El modelo relacional de bases de datos,” *Boletín de Política Informática*, vol. 6, pp. 53–61, 2003.

[14] Microsoft, “Crear y administras relaciones en PowerBI,” 2024, [Online]. Available: https://learn.microsoft.com/es-es/power-bi/transform-model/desktop-create-and-manage-relationships

[15] IBM, “Primary and foreign keys,” 2023, [Online]. Available: https://www.ibm.com/docs/es/ida/9.1.2?topic=entities-primary-foreign-keys

[16] Microsoft, “Creación de una tabla de fechas,” 2024, [Online]. Available: https://learn.microsoft.com/es-es/training/modules/design-model-power-bi/3-date-table

|  |
| --- |
| Anexos / Eranskinak |

