# UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

**Colegio de Ciencias e Ingenierías**


# Aplicación de aprendizaje automático para el desarrollo de modelos de planificación de demanda


## Sebastián Andrés Garzón García

**Ingeniería en Alimentos**


Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Ingeniero en Alimentos


Quito, 4 de diciembre de 2024

# UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

**Colegio de Ciencias e Ingenierías**

**HOJA DE CALIFICACIÓN**
**DE TRABAJO DE FIN DE CARRERA**

**Aplicación de aprendizaje automático para el desarrollo de modelos de planificación de demanda**

**Sebastián Andrés Garzón García**

**Nombre del profesor, Título académico** **Danny Navarrete, MSc.**

Quito, 10 de diciembre de 2024

# © DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en la Ley Orgánica de Educación Superior del Ecuador.

Nombres y apellidos: Sebastián Andrés Garzón García

Código: 00216983

Cédula de identidad: 0926283433

Lugar y fecha: Quito, 10 de diciembre de 2024

# ACLARACIÓN PARA PUBLICACIÓN

**Nota:** El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en http://bit.ly/COPETheses.

# UNPUBLISHED DOCUMENT

**Note:** The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on http://bit.ly/COPETheses.

## RESUMEN

La planificación de la demanda es un desafío fundamental en mercados globales caracterizados por alta volatilidad e incertidumbre, especialmente en la industria alimentaria, donde la gestión de inventarios y la reducción de desperdicios son prioritarias. Este trabajo evalúa el uso de aprendizaje automático para mejorar la predicción de la demanda en una empresa panificadora de consumo masivo, abordando la necesidad de superar las limitaciones de métodos tradicionales. Se implementaron y compararon cuatro modelos: XGBoost, Máquinas de Soporte Vectorial (SVR), k-Vecinos Más Cercanos (kNNR) y Bosques Aleatorios (RF). XGBoost obtuvo el mejor desempeño con un RMSLE de 0.4641, destacándose por su capacidad para manejar relaciones complejas y datos faltantes. A pesar de su precisión, enfrenta limitaciones al generalizar nuevos clientes o productos, lo que plantea desafíos para su implementación.

Los resultados confirman que el aprendizaje automático mejora significativamente la precisión de las predicciones y permite prácticas sostenibles en la gestión de inventarios. Este enfoque no solo optimiza cadenas de suministro, sino que también promueve el desarrollo económico sostenible. Se recomienda futuras investigaciones para explorar mejoras en escalabilidad, optimización de datos y adaptaciones para PYMES que enfrenten restricciones de recursos.


**Palabras clave**: Planificación de la demanda, aprendizaje automático, gestión de datos, PYMES, cadenas de suministro.

## ABSTRACT

Demand planning in volatile global markets is essential for optimizing inventories and reducing waste, particularly in the food industry. This study evaluates the use of machine learning to improve demand forecasting at a baked goods producer, comparing four models: XGBoost, Support Vector Regression (SVR), k-Nearest Neighbors Regression (kNNR), and Random Forest (RF). XGBoost achieved the best performance with an RMSLE of 0.4641, demonstrating high accuracy and the ability to handle complex relationships and missing data, although it faces limitations when generalizing to new client or product IDs.

Machine learning showed significant advantages over traditional methods, enhancing prediction accuracy and promoting sustainable practices in inventory management. To maximize its impact, ensuring adequate data collection and exploring its implementation in small and medium-sized enterprises (SMEs) through accessible and adaptive solutions is recommended. This approach can optimize supply chains and contribute to sustainable economic development.


**Keywords**: Demand planning, machine learning, demand forecasting, SMEs, supply chain.

# TABLA DE CONTENIDOS

# ÍNDICE DE TABLAS

# ÍNDICE DE FIGURAS

# ÍNDICE DE ANEXOS

# INTRODUCCIÓN

En la actualidad, la planificación de la demanda se ha convertido en un desafío crucial para las organizaciones que operan en mercados globales caracterizados por una creciente volatilidad e incertidumbre (Pournader et al., 2021). La capacidad de prever de manera precisa las necesidades de consumo es fundamental para optimizar la gestión de inventarios, reducir desperdicios, garantizar la seguridad del suministro y, en última instancia, contribuir al bienestar social y económico de la población (Toorajipour et al., 2020). Los sistemas tradicionales de predicción a menudo fallan al enfrentarse a la complejidad derivada de factores globales, culturales, sociales, ambientales y económicos, lo que genera la necesidad de adoptar enfoques más avanzados, como los modelos de aprendizaje automático (Nguyen et al., 2022).

Las demandas contemporáneas no solo responden a las dinámicas del mercado, sino que también están profundamente influenciadas por cambios socioculturales, como las tendencias de consumo sostenible, la preferencia por productos orgánicos o locales, y las fluctuaciones en la salud pública y seguridad alimentaria (Fildes et al., 2019). El impacto del cambio climático y las presiones ambientales también alteran los patrones de producción y consumo, complicando aún más la planificación efectiva de la demanda (Aamer et al., 2020). Ante esta realidad, surge la necesidad de implementar métodos predictivos más robustos y precisos que integren variables complejas y permitan a las empresas adaptarse a un entorno en constante cambio.

El estado del arte en la predicción de demanda ha avanzado significativamente en los últimos años, particularmente con la integración de técnicas de aprendizaje automático (Kasula, 2017). Modelos como el XGBoost, la Regresión por Máquinas de Soporte Vectorial (SVR), la Regresión k-Vecinos Más Cercanos (kNNR) y los Bosques Aleatorios (RF) han demostrado ser capaces de identificar patrones complejos y manejar la alta dimensionalidad de los datos, mejorando

sustancialmente la precisión de las predicciones en comparación con métodos tradicionales (Spiliotis, 2022). Estos modelos se han aplicado con éxito en diversas industrias, permitiendo gestionar mejor la variabilidad de la demanda y optimizar la cadena de suministro (Spliotis, 2022). Sin embargo, el desafío sigue siendo adaptar estas herramientas a contextos específicos, considerando limitaciones geográficas, culturales y económicas (Barocas et al., 2023).

En particular, la industria alimentaria enfrenta presiones adicionales debido a la necesidad de garantizar la seguridad alimentaria y minimizar el desperdicio de alimentos, lo que tiene implicaciones tanto económicas como ambientales (Roy et al., 2023). Aproximadamente, 17% de los alimentos producidos a nivel mundial se desperdicia entre la distribución y los consumidores finales (United Nations Environmental Program, 2021), lo que subraya la importancia de una planificación de demanda precisa que ayude a mitigar este problema. Además, la seguridad alimentaria y el bienestar de las comunidades dependen en gran medida de la capacidad de las empresas para mantener un flujo constante de productos, adaptándose rápidamente a las fluctuaciones en la oferta y demanda (Aamer et al., 2020).

Desde una perspectiva económica, mejorar la precisión en la planificación de la demanda también puede generar ahorros significativos. La reducción de inventarios excesivos o faltantes no solo optimiza el uso de los recursos financieros, sino que también contribuye a una mayor eficiencia operativa y una mejor toma de decisiones a nivel organizacional (Lyu et al., 2020). En este sentido, los modelos de aprendizaje automático presentan una oportunidad para superar las limitaciones de los métodos tradicionales, al permitir una mayor flexibilidad y adaptabilidad frente a la complejidad del mercado (Spiliotis, 2022).

El objetivo general de este estudio es desarrollar y evaluar modelos de planificación de demanda basados en aprendizaje automático que permitan mejorar la precisión de las predicciones y

optimizar la gestión de inventarios. Para alcanzar este propósito, se establecieron varios objetivos específicos:

1. Analizar los datos históricos de demanda con el fin de identificar patrones y factores clave que influyen en la variabilidad de la demanda.
2. Ajustar e implementar cuatro modelos de aprendizaje automático —XGBoost, Máquinas de Soporte Vectorial (SVR), k-Vecinos Más Cercanos (kNNR) y Bosques Aleatorios (RF)— optimizando sus hiperparámetros y evaluando su precisión en función de los datos históricos.
3. Comparar el desempeño de estos modelos, tanto en términos de precisión de predicción como en su impacto sobre la gestión de inventarios, con el objetivo de recomendar el modelo más efectivo e integrarlo en el proceso de planificación de demanda de la empresa.

Con base en estos fundamentos, se espera que la implementación de técnicas avanzadas de aprendizaje automático no solo contribuya a mejorar la precisión en la predicción de la demanda, sino que también promueva prácticas sostenibles en la gestión de inventarios, alineadas con los objetivos globales de reducción de desperdicios y optimización de recursos. Este enfoque en la planificación de la demanda tiene el potencial de mejorar la resiliencia de las cadenas de suministro, garantizando un mayor bienestar para la sociedad y promoviendo un desarrollo económico más sostenible.

# METODOLOGÍA

## Revisión de literatura

La inteligencia artificial (IA) se ha convertido en un campo de estudio esencial en las últimas décadas, abarcando una amplia gama de técnicas orientadas a la creación de sistemas capaces de imitar o superar ciertas capacidades humanas, como el razonamiento, la percepción y la toma de decisiones (Van Wynsberghe, 2021). Dentro de esta área, el aprendizaje automático ocupa un lugar central, al proporcionar métodos estadísticos y computacionales que permiten a las máquinas aprender patrones a partir de datos sin ser explícitamente programadas para una tarea específica (Kühl et al., 2022). En otras palabras, el aprendizaje automático convierte los datos en conocimiento accionable, lo cual resulta particularmente útil en el ámbito de la predicción de la demanda, donde las relaciones entre variables pueden ser altamente complejas y variar con el tiempo (Kühl et al., 2022).

El aprendizaje automático, como subcampo de la IA, se caracteriza por utilizar algoritmos que aprenden de la experiencia, es decir, a partir de conjuntos de entrenamiento de datos (Kühl et al., 2022). Mientras que la IA es un concepto amplio que engloba cualquier técnica que permita a la máquina mostrar inteligencia, el ML se centra en el uso de técnicas computacionales y estadísticos para identificar patrones y realizar predicciones o clasificaciones (Sarker, 2021). Dentro del ML se distinguen dos grandes tareas: la regresión y la clasificación. Un modelo de regresión se emplea cuando el objetivo es predecir una variable continua, como podría ser la demanda de un producto en unidades diarias o mensuales (Maulud & Abdulazeez, 2020). En cambio, un modelo de clasificación se utiliza para asignar etiquetas o categorías discretas a una observación, por ejemplo, determinar si un cliente comprará o no un artículo (Abdullah & Abdulazeez, 2021).

En el contexto del aprendizaje automático, se han desarrollado diversos algoritmos capaces de abordar problemas de regresión con altos niveles de precisión y eficiencia. Entre los más representativos se encuentran Random Forest (RF), k-Nearest Neighbors Regressor (kNNR), XGBoost y Support Vector Regression (SVR) (Spliotis, 2022). El Random Forest es un método de ensamble basado en la combinación de múltiples árboles de decisión, lo cual permite una mayor robustez y generalización frente a datos complejos y ruido (Probst et al., 2019). Por su parte, el kNNR se fundamenta en el principio de vecindad, estimando el valor a predecir a partir de las observaciones más cercanas en el espacio de características (Yang & Shami, 2020). XGBoost es una técnica basada en gradient boosting que construye múltiples árboles secuencialmente, corrigiendo los errores de los árboles anteriores y logrando así una alta eficiencia computacional y precisión, muy apreciada en competencias de ciencia de datos (Wang et al., 2021). Finalmente, SVR, una variante de las Máquinas de Soporte Vectorial orientada a regresión busca encontrar una función que se ajuste a los datos dentro de un umbral de error y que minimice la complejidad del modelo, resultando en funciones muy flexibles capaces de capturar relaciones complejas (Smets et al., 2007).

La evaluación del rendimiento de un modelo de regresión se lleva a cabo mediante métricas apropiadas. Una de estas es la Raíz del Error Cuadrático Medio del Logaritmo (RMSLE, por sus siglas en inglés), útil en problemas donde el rango de valores de la variable objetivo puede variar en órdenes de magnitud, o cuando penalizar proporcionalmente los errores relativos es deseable (Weerts et al., 2020). La RMSLE se define como la raíz cuadrada del promedio del error cuadrático del logaritmo de las predicciones y los valores reales (Weerts et al., 2020). Siendo $y_i$ los valores observados y $\hat{y}_i$ las predicciones del modelo, la RMSLE se calcula como (Weerts et al., 2020):

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

Por ejemplo, si se dispone de datos reales de demanda y sus correspondientes predicciones, se puede transformar ambas series añadiendo 1 a cada valor para evitar problemas con ceros y luego aplicar el logaritmo natural. Posteriormente, se calcula la diferencia entre las predicciones logarítmicas y los valores observados logarítmicos, se elevan al cuadrado, se promedian y finalmente se toma la raíz cuadrada. Esta métrica es especialmente útil cuando se quiere penalizar más la subestimación de valores grandes que la sobrestimación de valores pequeños (Weerts et al., 2020).

Otra consideración importante en la construcción de modelos de aprendizaje automático es la selección y ajuste de hiperparámetros. Los hiperparámetros son aquellos parámetros configurados antes del proceso de entrenamiento, que no se aprenden directamente de los datos (Arnold et al., 2024). Estos controlan el comportamiento del algoritmo de aprendizaje, influyendo sobre la complejidad, la capacidad de generalización y la eficiencia del modelo (Arnold et al., 2024). Por ejemplo, en un modelo de Random Forest, el número de árboles y la profundidad máxima de cada árbol son hiperparámetros que deben ajustarse (Probst et al., 2019). La optimización de hiperparámetros es esencial, ya que una elección inadecuada puede conducir a modelos subóptimos, con bajo rendimiento de predicción o problemas de sobreajuste (Arnold et al., 2024).

Con el fin de encontrar la mejor combinación de hiperparámetros, es habitual dividir el conjunto de datos original en subconjuntos: uno de entrenamiento, otro de validación y, finalmente, otro de prueba. Esta partición permite entrenar el modelo sobre una parte de los datos y evaluar su

rendimiento en datos no vistos, evitando así el sobreajuste (Janiesch et al., 2021). Además, métodos como la validación cruzada (k-fold validation testing) son esenciales para una evaluación más robusta. La validación cruzada implica dividir el conjunto de datos en k subconjuntos o "folds"; se entrena el modelo sobre k-1 subconjuntos y se valida en el restante. Este proceso se repite k veces y el rendimiento se promedia (Gorriz et al., 2024). Cuando se combina este procedimiento con la búsqueda en malla (gridsearch), es decir, la evaluación sistemática de todas las combinaciones posibles de hiperparámetros, se obtiene un método poderoso para encontrar configuraciones óptimas del modelo (Adnan et al., 2022).

Un aspecto fundamental del aprendizaje automático es su capacidad para capturar relaciones no lineales entre las variables (Baumer et al., 2021). A diferencia de los métodos estadísticos clásicos, que a menudo suponen linealidad entre predictores y respuesta, los algoritmos modernos como RF o XGBoost pueden modelar relaciones complejas y no lineales, adaptándose mejor a escenarios reales donde la relación entre las variables no sigue un patrón lineal simple (Baumer et al., 2021). Esta capacidad resulta particularmente valiosa en la predicción de la demanda, donde factores estacionales, tendencias y efectos promocionales pueden interactuar de forma no lineal (Baumer et al., 2021).

Por último, cabe mencionar el concepto de rezagos (lags). En un contexto de series temporales o pronóstico de demanda, un rezago se refiere a la inclusión de valores pasados de la misma variable objetivo o de variables exógenas como predictores del valor actual (Montgomery et al., 2015). Por ejemplo, el consumo de un producto en el día de hoy puede depender del consumo de ayer, de la semana pasada o del mismo día del mes anterior. Incluir estos rezagos permite a los modelos

capturar patrones temporales y dependencias dinámicas, lo que incrementa su capacidad de predicción a lo largo del tiempo (Montgomery et al., 2015).

## Comprensión del Negocio

El objetivo principal de este proyecto es predecir la demanda de productos para optimizar la gestión de inventarios y mejorar la planificación logística. Al estimar con precisión la cantidad de productos que cada cliente demandará en futuras semanas, la empresa puede gestionar de manera más eficiente su inventario, reducir costos operativos y aumentar la satisfacción del cliente (Farasyn et al., 2011). Esta predicción es fundamental para tomar decisiones informadas que impactan directamente en la eficiencia y rentabilidad del negocio. La optimización de inventarios permite minimizar el exceso o la falta de stock, reduciendo costos asociados a almacenamiento y pérdida de ventas (Farasyn et al., 2011). Además, una mejor planificación logística mejora la asignación de recursos y rutas de distribución basándose en la demanda prevista, lo que incrementa la eficiencia operativa y la satisfacción del cliente (Farasyn et al., 2011).

## Comprensión de los Datos

En esta etapa, se realizó una carga exhaustiva de los conjuntos de datos disponibles y se exploraron sus características para entender su estructura y contenido. Los conjuntos de datos principales incluyen *train.csv*, que contiene datos históricos de ventas. Además, se incorporaron tablas auxiliares como *cliente_tabla.csv*, *producto_tabla.csv* y *town_state.csv* para enriquecer el conjunto de características con información adicional sobre clientes, productos y ubicación geográfica de las agencias. Un diccionario con cada una de las tablas y sus variables se puede encontrar en los anexos.

Se seleccionaron columnas relevantes (Semana, Agencia_ID, Canal_ID, Ruta_SAK, Cliente_ID, Producto_ID, Demanda_uni_equil) para optimizar el uso de memoria y enfocarse en las variables más pertinentes para la predicción. Se implementó la función reduce_memory_usage para reducir el consumo de memoria de los DataFrames cargados, permitiendo manejar eficientemente grandes volúmenes de datos (Arora, 2024). Además, se realizó una unión de los conjuntos de datos principales con las tablas auxiliares mediante operaciones de merge, lo que permitió integrar información adicional de clientes, productos y ubicaciones. Se manejaron los valores faltantes rellenándolos con valores predeterminados ('Unknown') para variables categóricas, asegurando así la integridad y consistencia de los datos para etapas posteriores.

**Preparación de los Datos**

La preparación de los datos incluyó varias actividades de ingeniería de características y transformación de datos para mejorar la calidad y relevancia de las variables utilizadas en el modelado. Se extrajo información adicional de la columna NombreProducto, como el peso, la unidad de medida y el número de piezas. Estas características derivadas permitieron calcular el peso por pieza, proporcionando una visión más detallada de los productos.

Se calcularon medias de la demanda por Producto_ID y Cliente_ID a partir del conjunto de entrenamiento y se mapearon estas medias tanto en el conjunto de entrenamiento como en el de prueba. Además, se crearon características de retraso (lags) que capturan la demanda de semanas anteriores (Lag1 a Lag4), lo que ayuda a modelar la dependencia temporal en la demanda (Montgomery et al., 2015). Las variables categóricas fueron codificadas utilizando TargetEncoder, que asigna valores basados en la media de la demanda, facilitando así su uso en modelos de aprendizaje automático (Pargent et al., 2022). Se aplicó una transformación logarítmica a la variable objetivo para manejar la asimetría en su distribución y mejorar la capacidad de los

modelos para capturar patrones en los datos (Montgomery et al., 2015). Dada la cantidad de datos, solo se tomó una muestra de cien mil filas de los 72 millones existentes para no sobrecargar el poder computacional disponible, pues según la literatura, es mejor tomar una muestra representativa de los datos que tomar todos los datos y gastar recursos computacionales (Chaudhuri et al., 1998).

Finalmente, se escalaron las características numéricas utilizando StandardScaler, una técnica que ajusta los datos para que cada característica tenga una media de 0 y una desviación estándar de 1, especialmente para modelos sensibles a la escala de los datos como SVR y kNN (Raschka et al., 2020), asegurando que todas las variables contribuyan de manera equitativa al proceso de modelado.

**Modelado**

En la fase de modelado, se seleccionaron y entrenaron múltiples modelos de machine learning para predecir la demanda de productos. Además, se optimizaron los hiperparámetros de cada modelo mediante GridSearchCV para mejorar su rendimiento predictivo. Los modelos seleccionados incluyeron Soporte Vectorial de Regresión (SVR), K-Nearest Neighbors Regressor (kNN), Random Forest Regressor y XGBoost Regressor. Los hiperparámetros fueron seleccionados por recomendaciones de la literatura.

A continuación, se detallan las combinaciones de hiperparámetros investigadas para cada modelo durante el proceso de GridSearchCV:

Tabla 1. Hiperparámetros probados en el GridSearch

| Modelo | Hiperparámetro | Valores Investigados |
|---|---|---|
| Support Vector Regression | kernel | 'rbf' |
| | C | 0.5, 1.0 |
| K-Nearest Neighbors Regressor (kNN) | n_neighbors | 5, 10 |
| Random Forest Regressor | n_estimators | 50, 100 |
| | max_depth | 10, None |
| XGBoost Regressor | n_estimators | 50, 100 |
| | max_depth | 6, 10 |
| | learning_rate | 0.1, 0.05 |

Fuente: SVR (Smets et al., 2007), kNNR (Yang & Shami, 2020), RF (Probst et al., 2019) y XGBoost (Wang et al., 2021)

Estas combinaciones permitieron explorar diferentes configuraciones de cada modelo para identificar aquellas que proporcionan el mejor rendimiento predictivo basado en la métrica RMSLE. Para cada modelo, se utilizó GridSearchCV con validación cruzada de 10 pliegues y la métrica de evaluación mencionada. Los mejores modelos resultantes se almacenaron para su posterior evaluación.

Todo el proceso de entrenamiento y optimización de modelos se realizó en la plataforma Google Colab, aprovechando una máquina virtual (VM) con 51 GB de memoria. Esta configuración proporcionó el poder computacional necesario para manejar grandes volúmenes de datos y entrenar modelos complejos de manera eficiente, reduciendo significativamente el tiempo de procesamiento y permitiendo iteraciones rápidas durante el desarrollo del modelo.

**Evaluación**

Después del entrenamiento, se evaluó el rendimiento de cada modelo utilizando varias métricas de evaluación y se compararon para seleccionar el mejor modelo para la tarea de predicción. Las métricas utilizadas incluyeron RMSLE (Root Mean Squared Log Error). Se creó una tabla resumen

con las métricas de rendimiento para comparar los diferentes modelos, permitiendo identificar cuál presentó el menor valor de RMSLE, lo que indica una mejor capacidad para predecir la demanda de manera precisa.

Además, se realizó un análisis de autocorrelación (ACF) de la demanda semanal para determinar la influencia de los lags en la predicción. Las visualizaciones generadas, como histogramas de la demanda original y su transformación logarítmica, matrices de correlación de características y gráficos de dispersión comparando las predicciones del mejor modelo con la demanda real, permitieron comprender mejor la relación entre las variables y detectar posibles patrones o anomalías en los datos.

**Despliegue**

En la etapa de despliegue, se implementó un dashboard interactivo utilizando Power BI para visualizar las predicciones de demanda por IDs de clientes y productos. Este dashboard facilita la toma de decisiones estratégicas basadas en los resultados del modelo predictivo. El proceso de creación del dashboard incluyó la importación de las predicciones generadas (submission.csv) junto con los datos de train.csv para relacionar las predicciones con los IDs de clientes y productos existentes en el conjunto de entrenamiento. Se diseñaron diversas visualizaciones, como mapas de calor de demanda por cliente y producto, gráficos de barras de demanda total por producto, gráficos de líneas de demanda a lo largo del tiempo y filtros interactivos que permiten a los usuarios explorar los datos de manera dinámica.

Además, se añadieron elementos interactivos como slicers y tooltips para mejorar la experiencia del usuario y permitir una exploración detallada de las predicciones. El dashboard fue publicado en Power BI Service, permitiendo su acceso y visualización a través de la web, y se configuraron

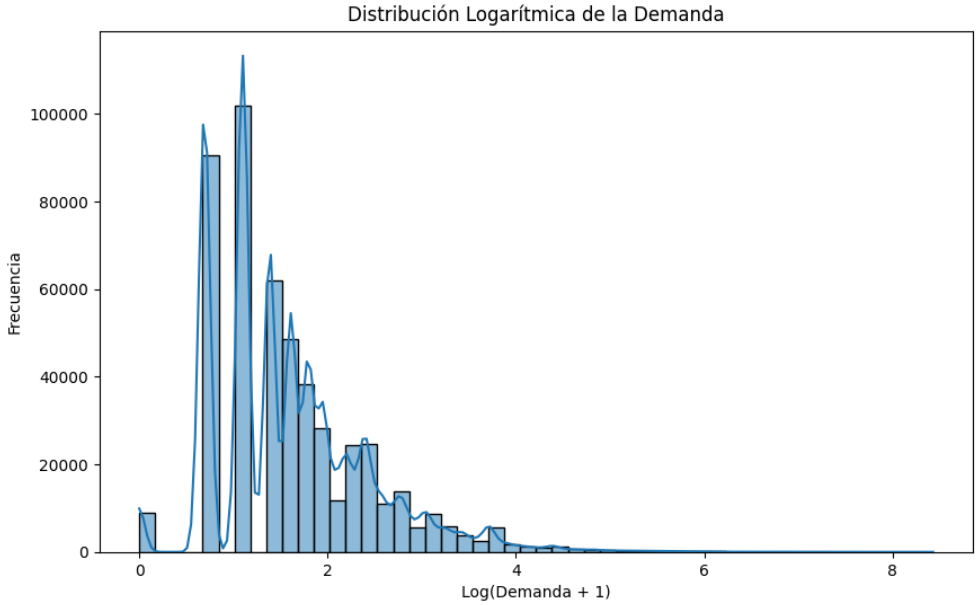
permisos de acceso para que los stakeholders clave pudieran interactuar con el dashboard y tomar decisiones informadas. La implementación de este dashboard en Power BI ofrece una visualización clara y concisa de las predicciones, facilita la exploración de datos en tiempo real y permite la integración con otras fuentes de datos y herramientas empresariales para una visión más completa.

# RESULTADOS Y DISCUSIONES

## Análisis descriptivo

El gráfico representa un histograma de la distribución logarítmica de la variable "Demanda", ajustada mediante la logarítmica, junto con una línea de densidad suavizada que muestra la frecuencia relativa de los datos. A primera vista, se observa que la distribución es asimétrica positiva, con una concentración marcada en los valores más bajos de la escala logarítmica y una cola extendida hacia la derecha. Esto indica que la mayoría de los datos representan demandas relativamente pequeñas, mientras que unos pocos valores corresponden a demandas altas en la escala original.

Figura 1: Distribución de Demanda_uni_equil




Fuente: Elaboración propia (2024)

El pico más alto del histograma, visible alrededor del valor logarítmico de 1, sugiere que la mayoría de las observaciones tienen una demanda ajustada cercana a este rango. Este valor en su forma

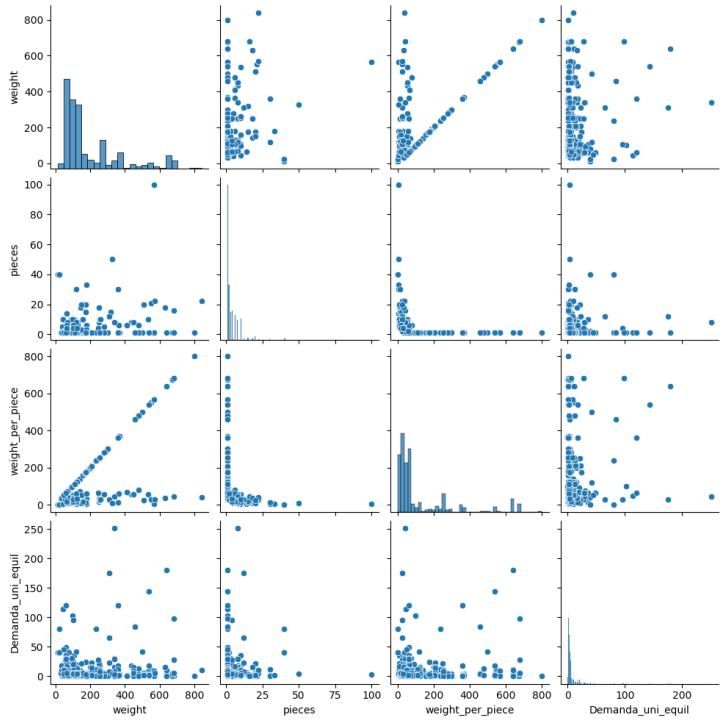
original correspondería aproximadamente a $e^{-1} - 1 \approx 1.72$, lo que refleja que la demanda más común en el conjunto de datos es relativamente baja. Este patrón es típico en conjuntos de datos de demanda, donde los productos más vendidos o las observaciones más frecuentes tienden a concentrarse en valores pequeños, mientras que los productos o eventos menos comunes pueden tener valores mucho mayores y contribuyen a la cola de la distribución.

La cola derecha del gráfico es extensa, alcanzando valores logarítmicos superiores a 6, lo cual equivale a demandas originales elevadas en algunos casos. Esto demuestra la existencia de observaciones atípicas que podrían tener un impacto significativo en el modelado predictivo. La transformación logarítmica aplicada redujo la asimetría y la amplitud de la dispersión en la escala original, haciendo que los datos sean más adecuados para modelos que asumen varianza constante, como regresiones lineales o aquellos que minimizan errores cuadráticos.

El análisis de la línea de densidad suavizada refuerza la idea de que la mayor concentración de los datos está en valores bajos, con una disminución gradual hacia la cola derecha. Esta suavidad es un indicativo positivo, ya que muestra una coherencia en la distribución de las observaciones (Wooldridge, 2013). Sin embargo, también subraya la necesidad de prestar atención a los valores extremos. La transformación logarítmica es útil en este sentido, ya que ayuda a estabilizar la varianza y mitigar el impacto de los outliers en los modelos de predicción (Wooldridge, 2013).

El gráfico de pares presentado ofrece una vista integral de las relaciones entre las variables numéricas weight, pieces, weight_per_piece, y Demanda_uni_equil. Este tipo de visualización es crucial para identificar patrones generales, correlaciones y distribuciones individuales, así como para detectar posibles anomalías que podrían influir en el análisis y modelado posterior.

Figura 2. Gráfico de correlación entre las variables de Producto_ID




Fuente: Elaboración propia (2024)

La distribución de la variable weight muestra una clara asimetría positiva, con la mayoría de los valores concentrados en pesos bajos, específicamente por debajo de 200, aunque algunos casos alcanzan valores extremos cercanos a 800. Esto indica la existencia de lotes pequeños predominantes en el conjunto de datos, pero también revela la presencia de productos o registros atípicos con pesos significativamente más altos. Por su parte, la variable de unidades por producto (pieces) también refleja una concentración en valores bajos, con una mayoría de observaciones cercanas a cero, mientras que algunos casos excepcionales involucran un mayor número de piezas. Esto refuerza la idea de que los lotes típicos contienen pocas unidades, mientras que lotes más grandes son poco frecuentes. La variable peso por pieza (weight_per_piece), como era de esperarse, exhibe una relación evidente con el peso total y el número de piezas, ya que está

derivada directamente de estas dos variables. En su distribución, se observa una asimetría similar, con valores predominantemente bajos y una dispersión significativa hacia rangos más altos.

Al analizar las relaciones entre las variables, se identifica un patrón entre weight y pieces, aunque esta no es estrictamente lineal. Los valores altos de peso suelen coincidir con un mayor número de piezas, lo que es consistente con la lógica esperada para productos que se venden en lotes. Sin embargo, también se observan puntos atípicos que presentan pesos elevados con pocas piezas, lo que podría corresponder a productos específicos o errores en los datos. Entre weight y weight_per_piece, la relación es casi lineal, ya que el peso total está directamente influido por el peso por unidad. En cambio, la relación entre pieces y weight_per_piece es inversa, lo que refleja que lotes con un mayor número de piezas suelen tener unidades más pequeñas, lo cual tiene sentido en un contexto práctico.

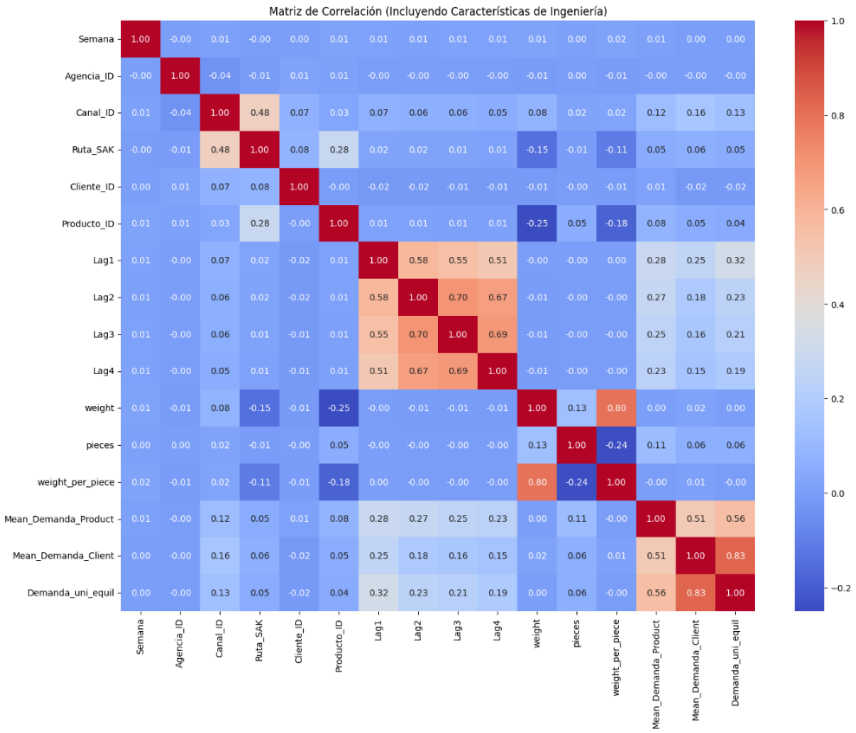
La variable Demanda_uni_equil muestra una dispersión considerable cuando se analiza en relación con las otras variables. Aunque no hay una correlación fuerte visible con weight, los valores de demanda más altos tienden a estar asociados a pesos y piezas moderados, mientras que las demandas extremas están más dispersas. Esto sugiere que la demanda podría depender de factores adicionales no representados en este análisis o que estas relaciones sean no lineales. Con respecto a pieces, se observa que las demandas más altas suelen estar relacionadas con lotes que contienen más piezas, aunque esta asociación no es estricta. Finalmente, no se identifica una relación evidente entre weight_per_piece y la demanda, lo que sugiere que el peso por pieza tiene poca influencia directa en el comportamiento de la variable objetivo.

El análisis general de este gráfico resalta varios aspectos importantes para el preprocesamiento y el modelado. Por un lado, las relaciones entre weight, pieces y weight_per_piece apuntan a una posible multicolinealidad que debería ser gestionada mediante la eliminación de variables

redundantes o la creación de combinaciones que resuman sus interacciones. Por otro lado, las distribuciones sesgadas de las variables y la presencia de valores extremos sugieren la necesidad de aplicar transformaciones como logaritmos o de emplear técnicas robustas para minimizar su impacto. Las relaciones no lineales identificadas refuerzan la idoneidad de algoritmos como XGBoost o Random Forest, que pueden capturar con mayor precisión estas complejidades. Sin embargo, es crucial explorar en mayor detalle los valores atípicos y su impacto potencial en el análisis predictivo.

El *heatmap* de correlaciones presentado muestra las relaciones lineales entre las diferentes variables del conjunto de datos, incluidas características originales e ingenierizadas (aquellas variables derivadas de las variables del set de datos original). Este análisis visual permite identificar qué variables tienen mayor relación con la variable objetivo (Demanda_uni_equil) y evaluar posibles redundancias o interacciones que podrían influir en el modelado predictivo (Mukhiya & Ahmed, 2020).

Figura 3. Matriz de correlación con variables del modelo




Fuente: Elaboración propia (2024)

La variable Demanda_uni_equil muestra correlaciones moderadas con algunos lags, en particular con Lag1 (0.32), Lag2 (0.23), y Lag3 (0.21). Aunque estas correlaciones no son extremadamente altas, sugieren que los valores de demanda en semanas anteriores tienen cierta influencia sobre la demanda actual. Esto respalda la inclusión de los lags como variables de entrada en modelos de aprendizaje automático, ya que, aunque las correlaciones son bajas, pueden capturar información útil en interacciones más complejas que los modelos no lineales podrían aprovechar (Raschka et al., 2020).

Otro hallazgo importante es la fuerte correlación entre Mean_Demanda_Product y Demanda_uni_equil (0.56), lo que indica que el promedio histórico de la demanda por producto

es un predictor relevante. Este resultado tiene sentido, ya que los patrones de consumo tienden a ser consistentes a nivel de producto (Joshi et al., 2023). Asimismo, Mean_Demanda_Client también muestra una correlación significativa con Demanda_uni_equil (0.83), lo que resalta la importancia del comportamiento histórico del cliente como una característica predictiva clave (Joshi et al., 2023). Este fuerte vínculo sugiere que incorporar información específica del cliente es esencial para mejorar la precisión del modelo.

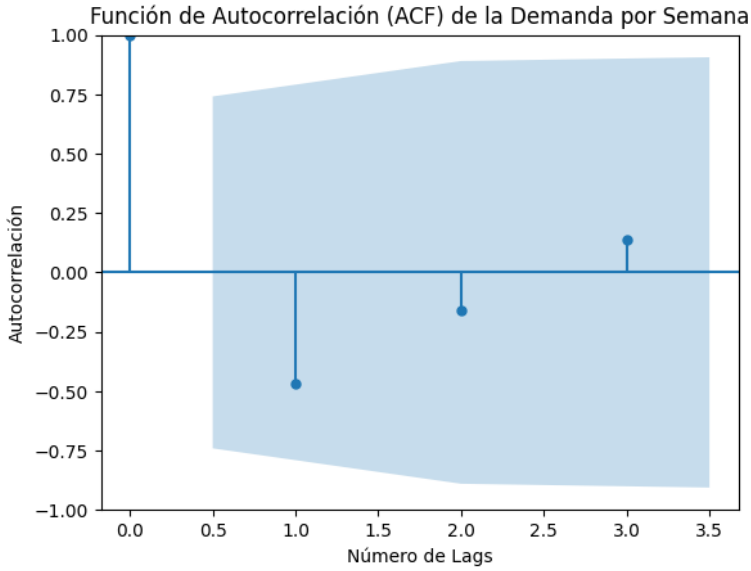
Las correlaciones entre los diferentes lags (por ejemplo, Lag1, Lag2, Lag3, y Lag4) son altas, con valores que van desde 0.55 a 0.70. Esto sugiere que existe una dependencia temporal entre estos valores, lo cual podría introducir multicolinealidad si se utilizan todas estas variables simultáneamente en modelos lineales. Sin embargo, para modelos de aprendizaje automático como XGBoost o Random Forest, esta multicolinealidad no es un problema crítico, aunque podría redundar en características que aporten información similar (Garg & Tai, 2013).

Por otro lado, variables como weight, pieces y weight_per_piece tienen correlaciones bajas o insignificantes con Demanda_uni_equil (-0.01 a 0.06). Esto sugiere que estas características, en su forma actual, aportan poco al modelo predictivo. Finalmente, cabe destacar que variables categóricas como Producto_ID, Cliente_ID, y Agencia_ID muestran correlaciones bajas con Demanda_uni_equil. Esto es esperado, ya que estas variables probablemente influyen de manera no lineal en la demanda y requieren transformaciones adicionales, como codificación categórica o embeddings, para que los modelos puedan capturar su impacto real (Bolikulov et al., 2024).

El gráfico muestra la función de autocorrelación (ACF) de la demanda por semana, donde se analizan los valores de correlación entre la serie temporal y sus propios rezagos (lags). Este análisis

es esencial para identificar patrones de dependencia temporal en los datos, lo que es fundamental al trabajar con series temporales y al modelar comportamientos dinámicos (Montgomery et al., 2015).

Figura 4. Gráfico de función de autocorrelación con la variable Demanda_uni_equil por semana




Fuente: Elaboración propia (2024)

En el gráfico, la autocorrelación en el lag 0 es, como es de esperarse, de 1.0, ya que cada observación está perfectamente correlacionada consigo misma (Montgomery et al., 2015). En los rezagos posteriores, el comportamiento cambia. En los lags 1, 2 y 3 la correlación es negativa y significativa, cayendo por debajo del límite del intervalo de confianza representado por las bandas azules. Este comportamiento indica que, bajo los parámetros del ACF, no existe una relación inversa entre la demanda de una semana y la de la semana siguiente (Montgomery et al., 2015). Es decir, si la demanda fue alta en una semana, estos novan a afectar las dinámicas de compra o consumo.

El hecho de que ningún lag sea superior o inferior al intervalo de confianza en el análisis de autocorrelación (ACF) indica que, desde un punto de vista estadístico tradicional, no hay una

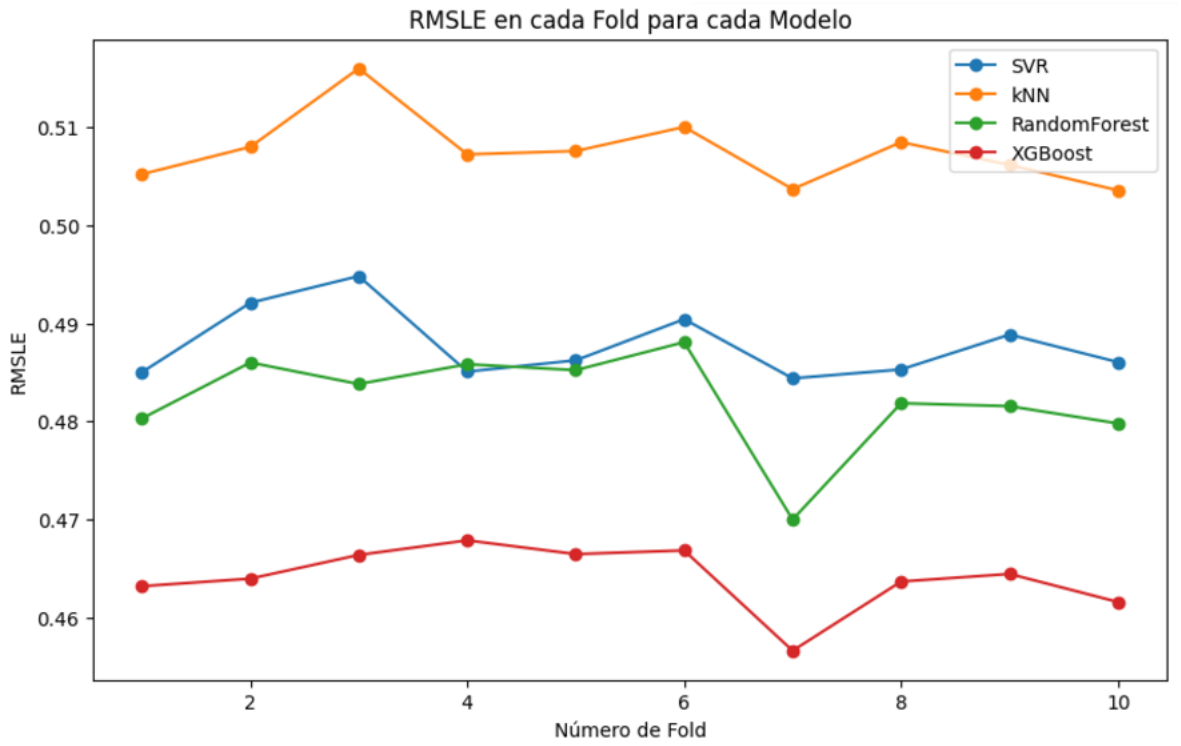
correlación significativa entre la demanda actual y los valores de demanda en semanas anteriores. Esto sugiere que, bajo un enfoque clásico de series temporales como ARIMA, estos lags podrían no tener un impacto suficientemente fuerte como para ser relevantes en el modelado (Montgomery et al., 2015). Sin embargo, esta observación no necesariamente implica que los lags sean irrelevantes en modelos de aprendizaje automático.

Por lo tanto, los lags podrían seguir siendo relevantes si contienen patrones latentes que los modelos puedan explotar. Por ejemplo, un modelo de aprendizaje automático podría detectar interacciones sutiles entre los valores de demanda en diferentes semanas que no se manifiestan claramente en el ACF pero que son útiles para mejorar la predicción.

**Resultados del entrenamiento**

Luego de realizar el entrenamiento de los modelos de aprendizaje automático con las variables explicadas en la metodología, se encontraron los siguientes resultados luego del k fold cross validation testing

Figura 5: Resultados promedio de los modelos luego del k fold validation testing




Fuente: Elaboración propia (2024)

Así mismo, se elaboró una tabla con los resultados promedios de cada uno de los modelos, siendo el modelo KGBoost Regression el que mejor puntaje obtuvo.

Tabla 2. Desempeño promedio de modelos de aprendizaje automático con set de validación

| Modelo | Desempeño promedio (RSMLE) |
|---|---|
| SVR | 0.4878 |
| RF | 0.4822 |
| XGBOOST | 0.4641 |
| kNNR | 0.5076 |

Fuente: Elaboración propia (2024)

De estas pruebas, se encontró que el algoritmo XGBoost es el que mejor desempeño tiene, pues es el que obtuvo el RSMLE menor entre los 4 modelos.

**Discusiones**

El análisis detallado del desempeño del modelo XGBoost ha confirmado que este es el algoritmo más efectivo en este caso, con un error cuadrático medio logarítmico (RMSLE) promedio de 0.4641. Esto lo coloca por encima de otros modelos evaluados, como Random Forest, SVR y kNNR. Es importante entender por qué XGBoost funcionó mejor que los otros algoritmos, lo que se debe a una combinación de sus características y cómo estas interactúan con el conjunto de datos utilizado.

XGBoost destaca por su capacidad para construir árboles de decisión de manera secuencial, donde cada nuevo árbol intenta corregir los errores cometidos por los árboles anteriores (Chen & Guestrin, 2016). Esto permite al modelo enfocarse en las áreas más problemáticas del conjunto de datos y ajustar continuamente sus predicciones; este enfoque, conocido como "boosting", es más eficiente que simplemente promediar múltiples árboles, como lo hace Random Forest (Chen & Guestrin, 2016). Por esta razón, XGBoost puede capturar patrones complejos que Random Forest, con su enfoque menos iterativo (Chen & Guestrin, 2016), no logra identificar con el mismo nivel de detalle.

Además, XGBoost tiene integradas herramientas para manejar problemas comunes en el aprendizaje automático, como el sobreajuste y la selección de características (Boehmke & Greenwell, 2019). Por ejemplo, utiliza regularización, una técnica que penaliza los modelos demasiado complejos y asegura que el modelo final no sea excesivamente dependiente de las particularidades del conjunto de entrenamiento (Boehmke & Greenwell, 2019). Esto es

particularmente útil cuando se trabaja con datos con muchas variables, como en este caso, donde las características incluyen IDs, promedios históricos y lags.

Otra ventaja clave de XGBoost es su capacidad para manejar relaciones no lineales entre las variables (Boehmke & Greenwell, 2019). A diferencia de modelos lineales como SVR, que asumen una relación directa y proporcional entre las características y el objetivo (Smets et al., 2007), XGBoost puede adaptarse a patrones más complejos y capturar interacciones entre variables que no son evidentes a simple vista (Boehmke & Greenwell, 2019). Esto le permitió, por ejemplo, aprovechar mejor las variables relacionadas con los lags y los promedios históricos, que tienen relaciones más complejas con la demanda.

En cuanto a por qué XGBoost superó a kNNR, esto se debe a que kNNR tiene dificultades para manejar conjuntos de datos con muchas observaciones y características complejas (Cosenza et al., 2021). Este modelo se basa en encontrar los puntos de datos más cercanos en el espacio de características, lo que puede volverse poco eficiente y menos preciso en datos de alta dimensionalidad como el que se utiliza aquí (Yang & Shami, 2020). Además, kNNR no tiene un mecanismo para priorizar características más importantes o manejar relaciones complejas entre ellas (Cosenza et al., 2021), lo que explica su peor desempeño.

En términos sencillos, XGBoost funcionó mejor porque está diseñado para aprender de manera eficiente, corregir errores de manera iterativa y adaptarse a patrones complejos. Esto lo hace ideal para datos como los analizados, que tienen interacciones no obvias entre las variables y requieren un modelo que pueda procesar grandes volúmenes de información sin perder precisión.

Es más, en comparación con el mejor modelo reportado en la literatura o por la competencia, que alcanza un RMSLE de 0.44, la diferencia es relativamente pequeña. Este margen indica que el

modelo actual es altamente competitivo y puede considerarse como una opción sólida para problemas similares, especialmente cuando se tiene en cuenta la complejidad inherente al conjunto de datos.

A pesar de su buen desempeño, una clara desventaja del modelo es su limitada capacidad para generalizar ante nuevos IDs de cliente o de producto que no estaban presentes en el conjunto de entrenamiento. Esto se debe a que el aprendizaje automático en XGBoost depende de patrones específicos asociados con las características de cada ID (Eckart et al., 2021), como Mean_Demanda_Client y Mean_Demanda_Product. Si se introduce un cliente o producto completamente nuevo, el modelo carecerá de la información histórica necesaria para generar predicciones precisas, lo que puede resultar en errores significativos en las primeras iteraciones.

Este problema puede abordarse de dos maneras. La primera consiste en aumentar el poder de cómputo disponible para entrenar el modelo con las 72 millones de filas completas del conjunto de datos. Esto permitiría que el modelo integre la mayor cantidad posible de variabilidad en los patrones de los IDs, reduciendo la probabilidad de encontrar situaciones inesperadas en los datos de prueba. No obstante, esta estrategia tiene una clara desventaja: requiere un acceso considerable a recursos computacionales avanzados, lo cual puede ser costoso, además de ser poco eficiente para escenarios en los que el conjunto de datos sigue creciendo de forma continua.

La segunda estrategia implica asegurarse de que los IDs más importantes, basados en su frecuencia o impacto en las predicciones, estén presentes en el conjunto de entrenamiento. Este enfoque reduce los requisitos computacionales y asegura que el modelo se ajuste bien a los elementos clave del negocio. Sin embargo, también puede introducir un sesgo en las predicciones, ya que los IDs menos representativos podrían no ser suficientemente considerados, afectando la capacidad del modelo para generalizar a situaciones nuevas o atípicas (Tsiamis et al., 2023).

Otra mejora que se podría considerar, aprovechando mayores recursos computacionales, es realizar un análisis más detallado de características mediante técnicas como feature engineering avanzada (Sahin, 2023). Por ejemplo, se podrían generar nuevas variables derivadas de las interacciones entre las existentes, como tendencias de demanda a lo largo del tiempo, segmentaciones específicas de clientes o productos, o valores estacionales ajustados. Este enfoque tiene el potencial de incrementar significativamente la capacidad predictiva del modelo. Asimismo, sería posible realizar una validación cruzada más robusta, probando configuraciones con un mayor número de folds o conjuntos de prueba más diversos, lo que incrementaría la robustez del modelo.

La viabilidad de utilizar esta tecnología en pequeños y medianos negocios (PYMES) plantea un escenario interesante. Aunque las PYMES no suelen contar con un equipo completo de analítica de datos, la implementación de modelos de aprendizaje automático puede ser una herramienta poderosa para optimizar operaciones, predecir demandas y reducir costos (Bauer et al., 2020). Este proceso podría iniciarse con la identificación de casos de uso concretos, como la previsión de ventas o el mantenimiento predictivo de maquinaria. La implementación en sí podría realizarse a través de plataformas en la nube o herramientas de software accesibles, que permitan entrenar, validar y desplegar modelos sin requerir una infraestructura tecnológica compleja, como los modelos de ERP que existen para empresas que recién empiezan (Jawad & Balázs, 2024). Asimismo, la alimentación de datos al modelo podría hacerse mediante la integración de fuentes internas (registros de ventas, inventarios, datos de clientes) y externas (información del mercado, precios de materias primas, tendencias de consumo), asegurando la correcta limpieza, normalización y actualización continua de la información (Jawad & Balázs, 2024).

Sin embargo, estas organizaciones deben tener en cuenta la importancia de invertir en sistemas de recolección de datos confiables. Sin una infraestructura adecuada para registrar y mantener datos relevantes, las ventajas del aprendizaje automático no podrán ser plenamente aprovechadas (Bauer et al., 2020). Por ello, las PYMES deben priorizar la adquisición de herramientas para la gestión y almacenamiento de datos como paso inicial antes de adoptar esta tecnología. En cuanto a si deben prestar atención a los pesos internos de las variables en el modelo, esto dependerá de los objetivos de la empresa. Si su propósito es comprender a fondo las razones detrás de las predicciones, resulta valioso analizar qué variables tienen mayor influencia en el resultado (Chen et al., 2020). Por el contrario, si la meta principal es la precisión del modelo, la interpretación de los pesos puede pasar a segundo plano, concentrándose en el desempeño general y las métricas clave (Chen et al., 2020). En cualquier caso, la elección entre optimizar la interpretabilidad o la precisión responderá a la estrategia del negocio y a la necesidad de entender el porqué de las decisiones automatizadas.

# CONCLUSIONES

El modelo XGBoost demostró ser el más efectivo para predecir la demanda en el conjunto de datos analizado, con un error cuadrático medio logarítmico (RMSLE) promedio de 0.4641. Su capacidad para capturar relaciones no lineales, manejar datos faltantes y ajustar los patrones históricos de demanda lo posiciona como una herramienta robusta y confiable frente a otros modelos evaluados, como Random Forest, kNN y SVR. Aunque no alcanzó el RMSLE de 0.44 reportado por el mejor modelo de la competencia, la diferencia es pequeña, lo que refleja su alto desempeño y viabilidad para implementaciones prácticas.

Una desventaja importante del modelo es su dependencia de los datos históricos específicos de clientes y productos. Esto implica que el modelo no generaliza correctamente cuando se enfrenta a IDs que no estaban presentes en el conjunto de entrenamiento, lo que podría limitar su aplicabilidad en escenarios dinámicos donde ingresan nuevos productos o clientes con frecuencia. Esta limitación señala la necesidad de abordar el problema mediante estrategias de recolección de datos y aumento de la capacidad computacional.

El uso de aprendizaje automático, particularmente con XGBoost, permitió superar las limitaciones inherentes de los modelos lineales al capturar interacciones complejas y no lineales entre las variables. Esto resultó en una mejora significativa en las predicciones, reflejando la ventaja de aplicar técnicas avanzadas de aprendizaje automático en problemas donde las relaciones entre los datos no son triviales. Este enfoque es especialmente relevante para industrias con alta variabilidad y grandes volúmenes de datos, como la predicción de demanda.

Se recomienda explorar cómo el aumento de la capacidad computacional puede mejorar el desempeño del modelo. Esto incluye entrenar con todo el conjunto de datos disponible (72

millones de filas), lo que permitiría capturar patrones más completos y mejorar la generalización. Adicionalmente, se podrían aplicar técnicas de optimización más avanzadas, como Bayesian optimization o el uso de herramientas distribuidas como Dask o Spark, para afinar los hiperparámetros y reducir el tiempo de cómputo.

Dado que el éxito de los modelos depende en gran medida de la calidad y amplitud de los datos, es crucial investigar estrategias para garantizar que los conjuntos de datos incluyan casos representativos de los IDs más relevantes. Esto podría implicar la implementación de sistemas de gestión de datos más avanzados, así como el diseño de políticas para priorizar la captura de información de nuevos productos o clientes desde el inicio de su ciclo de vida.

Dado que el modelo tiene dificultades para manejar nuevos IDs, futuras investigaciones podrían incorporar enfoques que incluyan predicciones con incertidumbre (como modelos bayesianos) para manejar mejor los escenarios inesperados. También sería valioso investigar técnicas para extrapolar información de IDs conocidos a nuevos, usando métodos de agrupación o embeddings para captar similitudes estructurales.

# REFERENCIAS BIBLIOGRÁFICAS

Aamer, A., Eka Yani, L., & Alan Priyatna, I. (2020). Data analytics in the supply chain management: Review of aprendizaje automático applications in demand forecasting. *Operations and Supply Chain Management: An International Journal*, *14*(1), 1-13.

Abdullah, D. M., & Abdulazeez, A. M. (2021). Machine Learning Applications based on SVM Classification A Review. *Qubahan Academic Journal*, *1*(2), 81–90. https://doi.org/10.48161/qaj.v1n2a50

Arora, I. (2024). Improving Performance of Data Science Applications in Python. *Indian Journal of Science and Technology*, *17*(24), 2499-2507.

Barocas, S., Hardt. M., & Narayanan, A. (2023). *Fairness and Machine Learning: Limitations and Opportunities*. The MIT Press.

Bauer, M., van Dinther, C., & Kiefer, D. (2020). Machine learning in SME: an empirical study on enablers and success factors.

Bolikulov, F., Nasimov, R., Rashidov, A., Akhmedov, F., & Young-Im, C. (2024). Effective Methods of Categorical Data Encoding for Artificial Intelligence Algorithms. *Mathematics*, *12*(16), 2553.

Chen, R., Dewi, C., Huang, S., & Caraka, R. E. (2020). Selecting critical features for data classification based on machine learning methods. *Journal of Big Data*, *7*(1). https://doi.org/10.1186/s40537-020-00327-4

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

Cosenza, D. N., Korhonen, L., Maltamo, M., Packalen, P., Strunk, J. L., Næsset, E., ... & Tomé, M. (2021). Comparison of linear regression, k-nearest neighbour and random forest methods in airborne laser-scanning-based prediction of growing stock. *Forestry: An International Journal of Forest Research*, *94*(2), 311-323.

Eckart, L., Eckart, S., & Enke, M. (2021). A brief comparative study of the potentialities and limitations of machine-learning algorithms and statistical techniques. In *E3S Web of Conferences* (Vol. 266, p. 02001). EDP Sciences.

Farasyn, I., Humair, S., Kahn, J. I., Neale, J. J., Rosen, O., Ruark, J., Tarlton, W., Van De Velde, W., Wegryn, G., & Willems, S. P. (2011). Inventory Optimization at Procter & Gamble: Achieving Real Benefits Through User Adoption of Inventory Tools. *INFORMS Journal on Applied Analytics*, *41*(1), 66–78. https://doi.org/10.1287/inte.1100.0546

Fildes, R., Ma, S., & Kolassa, S. (2019). Retail forecasting: Research and practice. *International Journal of Forecasting*, *38*(4), 1283–1318. https://doi.org/10.1016/j.ijforecast.2019.06.004

forecasting. *Operational Research*, *22*(3), 3037-3061.

Garg, A., & Tai, K. (2013). Comparison of statistical and machine learning methods in modelling of data with multicollinearity. *International Journal of Modelling, Identification and Control*, *18*(4), 295-312.

Gorriz, J. M., Clemente, R. M., Segovia, F., Ramirez, J., Ortiz, A., & Suckling, J. (2024, January 29). *Is K-fold cross validation the best model selection method for Machine Learning?* arXiv.org. https://arxiv.org/abs/2401.16407

Chaudhuri, S., Motwani, R., & Narasayya, V. (1998). Random sampling for histogram construction. *ACM SIGMOD Record*, *27*(2), 436–447. https://doi.org/10.1145/276305.276343

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, *31*(3), 685–695. https://doi.org/10.1007/s12525-021-00475-2

Jawad, Z. N., & Balázs, V. (2024). Machine learning-driven optimization of enterprise resource planning (ERP) systems: a comprehensive review. *Beni-Suef University Journal of Basic and Applied Sciences*, *13*(1). https://doi.org/10.1186/s43088-023-00460-y

Joshi, S. V., Shaikh, R. A. M. A., & Lohiya, L. J. (2023). *CONSUMER BEHAVIOUR: e-Book for MBA, 2nd Semester, SPPU*. Thakur Publication Private Limited.

Kasula, B. Y. (2017, August 17). *Machine Learning Unleashed: Innovations, applications, and impact across industries*. https://isjr.co.in/index.php/ITAI/article/view/169

Kühl, N., Schemmer, M., Goutier, M., & Satzger, G. (2022). Artificial intelligence and machine learning. *Electronic Markets*, *32*(4), 2235–2244. https://doi.org/10.1007/s12525-022-00598-0

Lyu, Z., Lin, P., Guo, D., & Huang, G. Q. (2020). Towards Zero-Warehousing Smart Manufacturing from Zero-Inventory Just-In-Time production. *Robotics and Computer-Integrated Manufacturing*, *64*, 101932. https://doi.org/10.1016/j.rcim.2020.101932

Maulud, D., & Abdulazeez, A. M. (2020). A review on Linear Regression comprehensive in Machine Learning. *Journal of Applied Science and Technology Trends*, *1*(2), 140–147. https://doi.org/10.38094/jastt1457

Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2015). *Introduction to Time Series Analysis and Forecasting*. John Wiley & Sons.

Mukhiya, S. K., & Ahmed, U. (2020). *Hands-On Exploratory Data Analysis with Python: Perform EDA techniques to understand, summarize, and investigate your data*. Packt Publishing Ltd.

Nguyen, T. T. H., Le, T. M., Bekrar, A., & Abed, M. (2022). Some Insights Into Effective Demand Planning. *IEEE Engineering Management Review*, *5*0(3), 141-148.

Pargent, F., Pfisterer, F., Thomas, J., & Bischl, B. (2022). Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features. *Computational Statistics*, *37*(5), 2671–2692. https://doi.org/10.1007/s00180-022-01207-6

Pournader, M., Ghaderi, H., Hassanzadegan, A., & Fahimnia, B. (2021). Artificial intelligence applications in supply chain management. *International Journal of Production Economics*, *241*, 108250. https://doi.org/10.1016/j.ijpe.2021.108250

Probst, P., Wright, M. N., & Boulesteix, A. L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: data mining and knowledge discovery*, *9*(3), e1301.

Raschka, S., Patterson, J., & Nolet, C. (2020). Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence. *Information*, *11*(4), 193. https://doi.org/10.3390/info11040193

Roy, P., Mohanty, A. K., Dick, P., & Misra, M. (2023). A review on the Challenges and Choices for Food Waste Valorization: Environmental and Economic Impacts. *ACS Environmental Au*, *3*(2), 58–75. https://doi.org/10.1021/acsenvironau.2c00050

Sahin, E. K. (2023). Implementation of free and open-source semi-automatic feature engineering tool in landslide susceptibility mapping using the machine-learning algorithms RF, SVM, and XGBoost. *Stochastic Environmental Research and Risk Assessment*, *37*(3), 1067-1092.

Sarker, I. H. (2021). Machine learning: algorithms, Real-World applications and research directions. *SN Computer Science*, *2*(3). https://doi.org/10.1007/s42979-021-00592-x

Smets, K., Verdonk, B., & Jordaan, E. M. (2007, August). Evaluation of performance measures for SVR hyperparameter selection. In *2007 International Joint Conference on Neural Networks* (pp. 637-642). IEEE.

Spiliotis, E., Makridakis, S., Semenoglou, A. A., & Assimakopoulos, V. (2022). Comparison of statistical and machine learning methods for daily SKU demand

Toorajipour, R., Sohrabpour, V., Nazarpour, A., Oghazi, P., & Fischl, M. (2020). Artificial intelligence in supply chain management: A systematic literature review. *Journal of Business Research*, *122*, 502–517. https://doi.org/10.1016/j.jbusres.2020.09.009

Tsiamis, A., Ziemann, I., Matni, N., & Pappas, G. J. (2023). Statistical learning theory for control: A finite-sample perspective. *IEEE Control Systems Magazine*, *43*(6), 67-97.

United Nations Environmental Program. (2021). Food Waste Index Report 2021. En https://www.unep.org/resources/report/unep-food-waste-index-report-2021.

Van Wynsberghe, A. (2021). Sustainable AI: AI for sustainability and the sustainability of AI. *AI And Ethics*, *1*(3), 213–218. https://doi.org/10.1007/s43681-021-00043-6

Wang, S., Zhuang, J., Zheng, J., Fan, H., Kong, J., & Zhan, J. (2021). Application of Bayesian hyperparameter optimized random forest and XGBoost model for landslide susceptibility mapping. *Frontiers in Earth Science*, *9*, 712240.

Weerts, H. J. P., Mueller, A. C., & Vanschoren, J. (2020, July 15). *Importance of tuning hyperparameters of machine learning algorithms*. arXiv.org. https://arxiv.org/abs/2007.07588

Wooldridge, J. M. (2013). *Introductory Econometrics: a Modern Approach*. Cengage Learning.

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295-316.

# ANEXOS

Anexo 1: Diccionario de la tabla train.csv

| Columna | Tipo de dato | Descripción |
|---|---|---|
| Semana | Entero | Número de la semana (de jueves a miércoles). |
| Agencia_ID | Entero | Identificador único del depósito de ventas. |
| Canal_ID | Entero | Identificador único del canal de ventas. |
| Ruta_SAK | Entero | Identificador único de la ruta (varias rutas corresponden a un depósito). |
| Cliente_ID | Entero | Identificador único del cliente. |
| Producto_ID | Entero | Identificador único del producto. |
| Venta_uni_hoy | Entero | Cantidad de unidades vendidas esta semana. |
| Venta_hoy | Decimal | Valor total de las ventas esta semana (en pesos). |
| Dev_uni_proxima | Entero | Cantidad de unidades devueltas la próxima semana. |
| Dev_proxima | Decimal | Valor total de las devoluciones la próxima semana (en pesos). |
| Demanda_uni_equil | Entero | Demanda ajustada del producto (variable objetivo a predecir). |

Fuente: Montoya, 2016

Anexo 2: Diccionario de la tabla cliente_tabla.csv

| Columna | Tipo de dato | Descripción |
|---|---|---|
| Cliente_ID | Entero | Identificador único del cliente. |
| NombreCliente | Cadena de texto | Nombre del cliente asociado al Cliente_ID. |

Fuente: Montoya, 2016

Anexo 3: Diccionario de la tabla producto_tabla.csv

| Columna | Tipo de dato | Descripción |
|---|---|---|
| Producto_ID | Entero | Identificador único del producto. |
| NombreProducto | Cadena de texto | Nombre del producto asociado al Producto_ID. |

Fuente: Montoya, 2016

Anexo 4: Diccionario de la tabla town_state.csv

| Columna | Tipo de dato | Descripción |
|---|---|---|
| Agencia_ID | Entero | Identificador único del depósito de ventas. |
| Town | Cadena de texto | Nombre de la localidad del depósito. |
| State | Cadena de texto | Estado al que pertenece el depósito. |

Fuente: Montoya, 2016

Anexo 5: Hiperparámetros implícitos de los cuatro modelos entrenados

| Modelo | Hiperparámetro | Descripción | Valor Predeterminado |
|---|---|---|---|
| SVR | degree | Grado del polinomio (aplicable solo para kernel 'poly'). | 3 |
| | gamma | Coeficiente para el kernel 'rbf', 'poly' y 'sigmoid'. | 'scale' |
| | coef0 | Término independiente en los kernels 'poly' y 'sigmoid'. | 0 |
| | tol | Tolerancia para los criterios de parada. | 0.001 |
| | shrinking | Si se usa la heurística de reducción del tamaño del modelo. | TRUE |
| | cache_size | Tamaño de la memoria caché (MB). | 200 |
| | verbose | Si se imprimen mensajes durante el ajuste. | FALSE |
| | max_iter | Número máximo de iteraciones. -1 significa sin límite. | -1 |
| KNeighborsRegressor | weights | Cómo se ponderan las contribuciones de los vecinos. | 'uniform' |
| | algorithm | Algoritmo utilizado para encontrar vecinos: 'auto', 'ball_tree', 'kd_tree', 'brute'. | 'auto' |
| | leaf_size | Tamaño de los nodos hoja en | 30 |

| Modelo | Hiperparámetro | Descripción | Valor Predeterminado |
|---|---|---|---|
| | | las estructuras de árboles. | |
| | p | Potencia de la métrica de Minkowski (p=2 es Euclidiana). | 2 |
| | metric | Métrica para calcular la distancia entre puntos. | 'minkowski' |
| | metric_params | Parámetros adicionales para la métrica seleccionada. | None |
| | n_jobs | Número de procesadores para la búsqueda de vecinos (-1 usa todos). | None |
| RandomForestRegressor | criterion | Función para medir la calidad de la división ('squared_error' o 'absolute_error'). | 'squared_error' |
| | min_samples_split | Número mínimo de muestras necesarias para dividir un nodo. | 2 |
| | min_samples_leaf | Mínimo número de muestras requeridas en una hoja. | 1 |
| | min_weight_fraction_leaf | Fracción mínima de peso de muestras requerida en una hoja. | 0 |
| | max_leaf_nodes | Número máximo de nodos hoja en el árbol. | None |
| | min_impurity_decrease | Umbral para realizar una división si | 0 |

| Modelo | Hiperparámetro | Descripción | Valor Predeterminado |
|---|---|---|---|
| | | mejora la pureza del nodo. | |
| | bootstrap | Si se utilizan muestras con reemplazo para construir los árboles. | TRUE |
| | oob_score | Si se calcula la puntuación fuera de la bolsa (Out-of-Bag). | FALSE |
| | n_jobs | Número de procesadores usados para el entrenamiento (-1 usa todos). | None |
| | random_state | Semilla para reproducibilidad. | None |
| | verbose | Nivel de verbosidad. | 0 |
| | warm_start | Si se reutilizan soluciones previas para ajustar más árboles. | FALSE |
| XGBoost | gamma | Reducción mínima en la pérdida requerida para dividir un nodo. | 0 |
| | min_child_weight | Peso mínimo de suma de las hojas para dividir un nodo. | 1 |
| | subsample | Fracción de muestras utilizadas para entrenar cada árbol. | 1 |
| | colsample_bytree | Fracción de características seleccionadas por árbol. | 1 |

| Modelo | Hiperparámetro | Descripción | Valor Predeterminado |
|---|---|---|---|
| | colsample_bylevel | Fracción de características por nivel de división. | 1 |
| | colsample_bynode | Fracción de características en cada nodo. | 1 |
| | reg_alpha | Regularización L1 (Lasso). | 0 |
| | reg_lambda | Regularización L2 (Ridge). | 1 |
| | scale_pos_weight | Balanceo para clases desbalanceadas. | 1 |
| | booster | Tipo de modelo base: 'gbtree', 'gblinear', o 'dart'. | 'gbtree' |
| | random_state | Semilla para reproducibilidad. | 0 |

Fuente: Elaboración propia