Optimización de la gestión de inventarios mediante análisis de ventas y modelado predictivo con Python utilizando una base de datos de Amazon del año 2022

Anyela Maria Mejia Nuñez

Asesor

Andrés Felipe Hernández Giraldo

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería ECBTI

Especialización Ciencia de Datos y Analítica

2025

Resumen

Para cualquier empresa el análisis de ventas es una herramienta fundamental para su crecimiento y rentabilidad, dado que ayuda a entender el comportamiento del mercado, descubrir nuevas oportunidades, mejorar la gestión del inventario y aumentar la competitividad en el mercado (Parkash, 2023). Amazon es una plataforma de comercio electrónico líder a nivel mundial, por lo tanto, analizar sus ventas puede ofrecer información relevante sobre el comportamiento del mercado. Este proyecto tiene como objetivo analizar y predecir el comportamiento de ventas para la optimización del inventario, utilizando una base de datos de Amazon correspondiente a los meses de abril, mayo y junio de 2022. A través del lenguaje de programación Python y bibliotecas como NumPy, Pandas, Scikit-learn, Matplotlib y Seaborn, se realizó un análisis exploratorio de datos, identificación de productos más vendidos y aplicación de modelos predictivos. Se evaluaron distintos algoritmos de clasificación y regresión, destacándose el modelo Random Forest como el más efectivo para predecir la variable “Category”, gracias a su capacidad para manejar el desbalance de clases y su alto rendimiento general. Para la variable “Size_Agrupada”, el modelo K-NN fue el mejor clasificador, logrando el mayor desempeño en términos de precisión general. En cuanto a la variable numérica “Qty_Mensual”, el modelo lineal obtuvo el mejor coeficiente de determinación (R^2), posicionándose como la mejor alternativa según el objetivo del análisis. El desarrollo del proyecto se enmarcó en la metodología CRISP-DM, permitiendo una estructura clara y replicable para futuros estudios.

Palabras claves: Análisis de venta, Python, análisis exploratorio de los datos, gestión de inventarios, CRISP-DM, modelo predictivo.

Abstract

For any company, sales analysis is a fundamental tool for its growth and profitability, since it helps to understand market behavior, discover new opportunities, improve inventory management and increase competitiveness in the market (Parkash, 2023). Amazon is a leading global e-commerce platform, therefore, analyzing its sales can provide relevant information about market behavior. This project aims to analyze and predict sales behavior for inventory optimization, using an Amazon database for the months of April, May and June 2022. Through the Python programming language and libraries such as NumPy, Pandas, Scikit-learn, Matplotlib and Seaborn, an exploratory data analysis, identification of best-selling products and application of predictive models were performed. Different classification and regression algorithms were evaluated, with the Random Forest model standing out as the most effective for predicting the "Category" variable, thanks to its ability to handle class imbalance and its high overall performance. For the variable "Size_Grouped", the K-NN model was the best classifier, achieving the highest performance in terms of overall accuracy. As for the numerical variable "Qty_Monthly," the linear model obtained the best coefficient of determination (R^2), positioning itself as the best alternative according to the objective of the analysis. The development of the project was framed within the CRISP-DM methodology, allowing a clear and replicable structure for future studies.

Keywords: Sales analysis, Python, exploratory data analysis, inventory management, CRISP-DM, predictive model.

Tabla de Contenido

Lista de Tablas

Lista de Figuras

Lista de Apéndices

Planteamiento del Problema

En el entorno competitivo actual, encontramos empresas que venden productos a través de plataformas de comercio electrónico o tiendas físicas que enfrentan desafíos significativos en la gestión de inventarios y en la toma de decisiones estratégicas (Llatas Fernandez, 2024). Según Llatas Fernandez (2024), en su trabajo hacen referencia a un reporte de "The Economist Intelligence Unit (2023)", donde dice que las organizaciones europeas y norteamericanas han visto una reducción anual en su rentabilidad de entre 7% y 9% como resultado de una deficiente gestión de inventarios. Una posible causa, es la inadecuada planificación de la demanda y de los patrones de ventas que conducen a problemas como el exceso o la falta de inventario, lo cual afecta negativamente la operatividad y los resultados financieros (Yobel SCM Corp, 2024). Es entonces la importancia de una buena gestión del inventario para las empresas, dado que un buen manejo del mismo ayuda a mejorar la eficiencia, la satisfacción del cliente y la rentabilidad del negocio (Paz Soldan Flores & Sánchez Farfán, 2024).

Para ello, es crucial implementar herramientas y modelos predictivos que permitan analizar los datos de venta de manera efectiva para tomar decisiones basadas en la información analizada. Según Cevallos Guamán et al. (2024) "la implementación de un modelo predictivo tiene el potencial de reducir los costos de inventario en un estimado del 12%, basado en la capacidad del modelo de prever con precisión la demanda futura".

Actualmente, los entornos de las empresas cada vez son más dinámicos y competitivos, el análisis de ventas es una herramienta de alta relevancia para las empresas que desean optimizar sus procesos alrededor de las ventas y la gestión de inventario, así como también la necesidad de predecir y planificar con mayor certeza las operaciones y las ventas se ha convertido en un elemento clave para el crecimiento de las empresas (Chicaiza et al., 2024).

La conjugación y aplicación de herramientas de análisis de datos y modelos predictivos son esenciales para el desarrollo de este trabajo, en el libro llamado Métodos Avanzados para Ventas y Operaciones manifiestan que: “En la gestión predictiva, se aplican técnicas de análisis de datos, aprendizaje automático, estadísticas y modelado matemático para identificar patrones, relaciones y oportunidades ocultas en los datos. Esto permite a las organizaciones anticipar cambios en la demanda del mercado, las necesidades de inventario, la capacidad de producción y otros aspectos críticos de sus operaciones comerciales” (Chicaiza et al., 2024, p. 19).

Indistintamente del sector o industria, abordar el problema de una inadecuada gestión de inventarios es crucial para mejorar la eficiencia y efectividad de las empresas. La implementación de modelos predictivos basados en el análisis de datos no solo optimiza dicha gestión, sino que también mejora la experiencia del cliente y fortalece la posición competitiva en el mercado. Si este problema persiste, las empresas se enfrentan a riesgos como pérdidas financieras significativas, disminución de su competitividad y una menor satisfacción del cliente (Yi, 2023).

En consecuencia, la predicción de ventas mediante modelos de aprendizaje automático se ha convertido en una herramienta estratégica para mejorar la eficiencia operativa y la planificación en entornos altamente competitivos. No obstante, el desarrollo de modelos predictivos efectivos requiere de un proceso sistemático que incluya limpieza de datos, selección de variables, entrenamiento y validación, todo ello adaptado al contexto específico de cada negocio. Además, en un marco más amplio, estas soluciones basadas en datos promueven la innovación tecnológica y la adopción de prácticas de toma de decisiones más informadas y estratégicas.

En este sentido, el presente trabajo busca abordar esta necesidad mediante la aplicación del ciclo CRISP-DM y la construcción de modelos de clasificación y regresión utilizando la base de datos pública de Github llamada “Amazon Sales Analysis” (Leon Jose, 2023). El objetivo es predecir variables clave relacionadas con el comportamiento de venta de los productos, como la cantidad de ítems vendidos y los ingresos generados. De esta manera, se espera demostrar el valor que la ciencia de datos puede ofrecer para optimizar la gestión comercial en plataformas de comercio electrónico.

Justificación

Este proyecto tiene relevancia y pertinencia en función de la necesidad de mejorar la gestión de inventarios y predecir las ventas para la toma de decisiones estratégicas de cualquier empresa. En el artículo de Cevallos Guamán et al. (2024) hacen referencia a un caso de estudio de la empresa Walmart, donde gracias a la implementación del Big Data y la analítica predictiva lograron reducir el desabastecimiento en un 16% ajustando los niveles de inventario en tiempo real y aumentar las ventas en un 4% prediciendo la demanda.

Es importante resaltar los múltiples beneficios de la implementación de un proyecto de análisis de ventas y desarrollo de modelos predictivos para las empresas como son:

1. Optimización de inventarios: Mediante la identificación de patrones y tendencias en los datos de ventas, se pueden predecir con mayor precisión las futuras demandas de productos. Esto permite optimizar la gestión de inventarios, evitando tanto el exceso como la falta de stock, asignación de recursos de manera eficiente evitando desperdicios, reducción de gastos innecesarios, entre otros aspectos. Adicionalmente, al asegurar la disponibilidad de productos y prevenir faltantes, se mejora la experiencia de compra del cliente, lo cual puede resultar en una mayor satisfacción y fidelización (Chicaiza et al., 2024).

2. Toma de decisiones estratégicas: Los responsables de la toma de decisiones pueden basarse en pronósticos, datos y modelos confiables para planificar y ejecutar estrategias de gestión de inventario, marketing y publicidad, distribución, finanzas, entre otras, de una manera más rápida y precisa, con el fin que proporcionen reducción de costos y una ventaja competitiva en el mercado. (Rey Escobar & Valle Nieto, 2024). Ejemplo, en una industria, se pueden tomar decisiones estratégicas sobre la producción y abastecimiento en función de la previsión de ventas y análisis de los patrones históricos de demanda, regulando la asignación de

recursos como mano de obra y materiales, minimizando los costos de almacenamiento y transporte, y garantizando que los productos a producir estén disponibles cuando los clientes los necesiten, aumentando su satisfacción y fidelidad (Chicaiza et al., 2024). Otro ejemplo, es cuando una empresa de comercio electrónico decide analizar los atributos de los clientes para mostrar anuncios específicos que mejoren las ventas (Subhendu Kumar Pani et al., 2019).

3. Competitividad en el Mercado: En un entorno altamente competitivo, contar con herramientas avanzadas de análisis y predicción de ventas puede otorgar una ventaja competitiva significativa. Las empresas pueden anticiparse a las tendencias del mercado y adaptar sus estrategias en consecuencia.

4. Uso de técnicas de ciencia de datos: al utilizar técnicas avanzadas de ciencia de datos y aprendizaje automático, se impulsa la innovación y el desarrollo tecnológico. Fomentar el uso de estas herramientas ayuda a construir una cultura organizacional orientada a la innovación y la mejora continua.

A nivel personal representa una valiosa oportunidad para mi desarrollo profesional, ya que permite aplicar técnicas de ciencia de datos y aprendizaje automático. La implementación de modelos predictivos y el análisis de datos de ventas facilitan la aplicación de conocimientos teóricos y metodológicos adquiridos durante la especialización.

Este trabajo cobra especial relevancia al aplicar modelos de clasificación y regresión sobre datos reales de una plataforma de comercio electrónico reconocida, abordando variables estratégicas como categoría, talla y cantidad. Además de fortalecer habilidades prácticas en el uso de metodologías como CRISP-DM, el proyecto aporta un análisis técnico replicable por profesionales y organizaciones interesadas en integrar la analítica predictiva en sus procesos de toma de decisiones.

Objetivos

Objetivo General

Optimizar la gestión de inventarios a través del análisis de ventas y la aplicación de modelos predictivos con Python, utilizando datos de Amazon del año 2022, para identificar patrones de compra y apoyar la toma de decisiones estratégicas.

Objetivos Específicos

Transformar los datos de venta mediante técnicas de preprocesamiento con OpenRefine y Python, asegurando su calidad, consistencia y utilidad para el desarrollo del análisis exploratorio y el modelado predictivo.

Explorar los datos utilizando herramientas de análisis y visualización en Python para identificar patrones, correlaciones y tendencias que influyen en las ventas.

Evaluar distintos modelos de aprendizaje automático en Python, aplicando validación cruzada y métricas de desempeño para seleccionar el modelo más adecuado en la predicción de ventas y optimización del inventario.

Marco Conceptual

- Accuracy: “representa el porcentaje total de valores correctamente clasificados, tanto positivos como negativos.” (Roberto Díaz Badra, 2020).
- Amazon: “Amazon es una multinacional tecnológica estadounidense especializada en comercio electrónico, computación en la nube, streaming digital e inteligencia artificial. Fundada en 1994 por Jeff Bezos, Amazon es uno de los mayores minoristas en línea del mundo, y ofrece una gama de productos que van desde libros y música hasta muebles y ropa” (Arimetrics, s. f.).
- Análisis de datos: “Es el proceso de exploración, transformación y examinación de datos para identificar tendencias y patrones que revelen insights importantes y aumenten la eficiencia para respaldar la toma de decisiones.” (Alteryx, s. f.).
- Análisis descriptivo: “Es una parte del proceso de analítica más amplio que incluye la analítica predictiva y prescriptiva. Mediante una visión integrada del rendimiento y las tendencias, como el crecimiento de las ventas año tras año o la cantidad de clientes, la analítica descriptiva permite que en las empresas se identifiquen las áreas de fortalezas y debilidades a fin de guiar las decisiones y la estrategia.” (Alteryx, s. f.).
- Análisis de venta: Es el proceso de examinar los datos de transacciones comerciales para comprender el comportamiento de los consumidores, identificar tendencias y patrones, y evaluar el desempeño de los productos. Este análisis permite a las empresas tomar decisiones informadas sobre marketing, inventarios y estrategias de precios.
- Árboles de decisión: “Modelos de representación gráfica que utilizan una estructura de árbol para tomar decisiones basadas en condiciones específicas de las características” (Velasco Rebolledo, 2024).

- Aprendizaje automático: “El aprendizaje automático (o machine learning) es el proceso iterativo que utiliza una computadora para identificar patrones de un conjunto de datos luego de que se le proporcionan restricciones específicas. Implica “entrenar” a una computadora para que explore entornos y adquiera nuevas habilidades sin programarla explícitamente para ello.” (Alteryx, s. f.).

- Error absoluto medio (EMA): “Se trata de una medida de las discrepancias absolutas típicas entre los valores reales de un conjunto de datos y los valores proyectados” (*Regression Metrics*, 2024).

- Error cuadrático medio (MSE): “Es una métrica que se utiliza para medir la precisión o el ajuste de un modelo predictivo, especialmente cuando las predicciones son valores numéricos continuos. El MSE cuantifica en qué medida los valores predichos de un modelo se alinean con los valores reales observados en el conjunto de datos” (*Regression Metrics*, 2024).

- F1-score: “es una métrica muy utilizada en problemas en los que el conjunto de datos a analizar está desbalanceado, combina precision y recall, para obtener un valor mucho más objetivo.” (Roberto Díaz Badra, 2020).

- K-NN: “Es una de las técnicas de clasificación más simples, pero es de las más utilizadas. Clasifica los registros de acuerdo con el número de vecinos más cercanos a ellos según su distancia euclidiana. Un registro particular se asigna a la clase a la que pertenezca la mayoría de sus vecinos más cercanos” (Yajure Ramírez, 2022).

- Limpieza de datos: “También conocida como depuración, identifica y corrige errores, duplicados y datos irrelevantes de un conjunto de datos sin procesar. Como parte del proceso de preparación de datos, la limpieza de datos permite obtener datos precisos y

sustentables que generan visualizaciones, modelos y decisiones empresariales de confianza." (Alteryx, s. f.).

- Matplotlib: Es una librería que sirve para crear y personalizar gráficos de dos y tres dimensiones como: diagramas de barra, de línea, de caja, histogramas, entre otros (Amazon Web Services, s. f.).

- Metodología CRISP-DM: "La metodología CRISP-DM (Cross-Industry Standard Process for Data Mining) integra todas las tareas necesarias en los proyectos de minería de datos, desde la fase de comprensión del problema hasta la puesta en producción de sistemas automatizados analíticos, predictivos y/o prospectivos. Esta metodología se puede aplicar a una gran variedad de problemas tales como encontrar perfiles de clientes fraudulentos, estimar la probabilidad de que los clientes de una empresa se vayan a la competencia o también determinar patrones de compra para luego, recomendar productos de interés" (Daniel Álvarez Gil, 2021).

- NumPy: Es una librería que se usa para crear y administrar matrices con grandes volúmenes de datos y ejecutar operaciones matemáticas con facilidad (Amazon Web Services, s. f.).

- OpenRefine: es una herramienta gratuita y de código abierto, que se utiliza para limpiar y transformar grandes cantidades de datos en diferentes formatos (Artem Makarov & Dmitry Namiot, 2023).

- Pandas: Sirve para manejar y analizar datos estructurados en tablas y matrices, por ejemplo, se puede utilizar para visualizar, importar, escribir, combinar, filtrar y agrupar los datos (Amazon Web Services, s. f.).

- Precisión: "es utilizada para poder saber qué porcentaje de valores que se han clasificado como positivos son realmente positivos." (Roberto Díaz Badra, 2020)

- Pronóstico de demanda: “Estima la demanda futura de productos y servicios, lo que ayuda a tomar decisiones comerciales fundamentadas. Puedes mejorar su exactitud incorporando información en tiempo real, analítica avanzada, aprendizaje automático y ciencia de datos. Un pronóstico de la demanda preciso ayuda con la administración del inventario, la planificación de la capacidad, la demanda de productos y la asignación de recursos.” (Alteryx, s. f.).

- Puntuación R-cuadrado (R^2): “Es una métrica estadística que se utiliza con frecuencia para evaluar la bondad del ajuste de un modelo de regresión. Cuantifica el porcentaje de variación de la variable dependiente al que contribuyen las variables independientes del modelo. R2 es una estadística útil para evaluar la eficacia general y el poder explicativo de un modelo de regresión” (*Regression Metrics*, 2024).

- Python: Es un lenguaje de programación ampliamente utilizado en la ciencia de datos debido a su simplicidad y las poderosas bibliotecas disponibles (Amazon Web Services, s. f.).

- Raíz del error cuadrático medio (RMSE): “Mide la raíz cuadrada de las discrepancias promedio entre los valores reales y los valores proyectados de un conjunto de datos. El MSE se utiliza con frecuencia en problemas de regresión y se utiliza para evaluar el funcionamiento de los modelos predictivos” (*Regression Metrics*, 2024).

- Random Forest: es un algoritmo de aprendizaje automático, que combina el resultado de múltiples árboles de decisión para llegar a un resultado único. (*¿Qué es el bosque aleatorio?*, 2025).

- Recall: “es utilizada para saber cuántos valores positivos son correctamente clasificados.” (Roberto Díaz Badra, 2020).

- Regresión Lineal: “se utiliza para construir una ecuación matemática lineal para la relación entre una variable dependiente y variables independientes. La idea principal de este modelo es obtener una ecuación lineal que se ajuste mejor al conjunto de datos. La línea más apropiada es aquella en la que el error de la predicción total para todos los puntos de datos es lo más pequeño posible” (Yi, 2023).
- Regresión Logística: “Utilizada para problemas de clasificación binaria, la regresión logística estima la probabilidad de que una instancia pertenezca a una clase particular” (Velasco Rebolledo, 2024).
- Scikit-learn: Es una librería que sirve para realizar análisis predictivos, debido a que ofrece una amplia gama de algoritmos y herramientas que hacen más fácil la implementación y aplicación de técnicas de aprendizaje automático en distintos problemas.
- Seaborn: Es una librería que se usa para crear y visualizar gráficos estadísticos, Permitiendo explorar y comprender rápidamente los datos.
- Validación cruzada: “es un procedimiento de análisis estadístico utilizado para evaluar la efectividad de una técnica de aprendizaje automático, así como un método de remuestreo utilizado para validar un algoritmo si no hay datos suficientes” (Sai Nikhil Boyapati & Ramesh Mummidi, 2020).
- Visualización de datos: “Es la representación visual de datos mediante gráficos, diagramas o gráficos informativos. La visualización de datos es cada vez más interactiva, lo que permite a los espectadores interactuar con la representación para profundizar en el análisis y la comprensión o cambiar de perspectiva. Sea cual sea el medio, la visualización de datos sirve a un propósito importante: comunicar tendencias y oportunidades clave a las partes interesadas para una toma de decisiones eficaz, haciendo más accesible la información compleja.” (Alteryx, s. f.).

Marco Teórico

A continuación, se presenta una revisión de la literatura sobre análisis de datos, modelado predictivo, limpieza de datos, visualización de datos, entre otras variables, que permiten abordar el estado actual del conocimiento de nuestro problema a resolver.

Yajure Ramírez (2022), analiza los datos de facturación de energía eléctrica en Uruguay durante el periodo 2000-2022, utilizando algoritmos de aprendizaje automático. Primero hicieron el proceso de preparación de los daos, para luego realizar un análisis exploratorio de los mismos y generar los gráficos. Aplicaron un algoritmo K-Means para agrupar los datos de acuerdo con los tipos de clientes, generando un clúster y agrupando a la perfección. El algoritmo K-NN para generar un modelo que permitiera predecir el tipo de cliente si era residencial o no, con una exactitud del 100%. Analisis de correlación y análisis PCA para reducir la dimensionalidad. El algoritmo de Regresión lineal para obtener un modelo para predecir la energía eléctrica total facturada de los nuevos clientes, obteniendo un coeficiente de determinación R2 de 0,981.

Cevallos Guamán et al. (2024), tuvo como objetivo examinar la implementación de Big Data y evaluar el impacto de un modelo de Analítica Predictiva en la precisión de previsiones de ventas y optimización de inventario, en una empresa minorista en Ecuador llamada "Or Importaciones", utilizaron una metodología mixta, combinando técnicas cuantitativas y cualitativas, además de la metodología CRISP-DM. El estudio, realizado entre enero y mayo de 2024, empleó Python y Google Colab para el análisis exploratorio de los datos. Primero entrenaron un modelo de Regresión Lineal el cual obtuvo un R2 de 0.85. Luego implementaron un modelo de Random Forest el cual obtuvo un R2 de 0.90. Por último, implementaron el modelo de Gradient Boosting, el cual dio mejores resultados mostrando una mejora significativa en la precisión de predicciones de ventas, con un R2 de 0.92.

Artem Makarov & Dmitry Namiot (2023), manifiestan la importancia del proceso de limpieza de los datos, muestran los principales métodos, analizan sus ventajas e inconvenientes y ofrece recomendaciones generales para la mejora de este proceso. Hace referencia a diferentes bibliotecas básicas como Pandas, NumPy y scikit-learn, y programas como OpenRefine.

Parkash (2023), analiza datos de venta de varios establecimientos de una empresa llamada “Big Mart”, con el fin de descubrir la correlación entre varios factores que influyen en las ventas y los ingresos. Primero se realiza la limpieza de los datos, se generan diferentes tipos de gráficos y aplica modelos de aprendizaje automático como: arboles de decisión el cual obtuvo un porcentaje de precisión de 62. El modelo de regresión lineal obtuvo una presión de 51. El modelo de bosque aleatorio obtuvo una presión de 64 y el modelo XG Boost obtuvo una presión de 69, el cual fue el mejor y con el que se analizaron finalmente los datos.

Subhendu Kumar Pani et al. (2019), detalla el análisis exploratorio de un conjunto de datos electrónicos de Amazon, utilizando Python para la implementación de diferentes técnicas de análisis. Lo primero fue la limpieza de los datos, la correlación de las variables, se graficaron y se realizó un análisis estadístico descriptivo. Además, muestran diferentes conceptos relaciones con el análisis de datos.

Ramachandra et al. (2021), tuvo como objetivo predecir las ventas de una empresa en "Black Friday". Proponiendo un modelo de regresión de bosque aleatorio, el cual dio una precisión promedio del 83,6%, esto se aplicó a un conjunto de datos de ventas de neumáticos del viernes negro. Primero realizaron el preprocesamiento de los datos, luego la correlación de las variables independientes con la variable objetivo y por último la implementaron del modelo para predecir las ventas.

Huo (2021), usa un conjunto de datos de venta de Walmart, con el fin de predecir las ventas diarias. Ponen a prueba 2 modelos lineales (método de suavizado exponencial y modelo ARIMA), 3 modelos de aprendizaje automático (algoritmo de regresión, XGBoost y Random Forest) y 2 modelos de aprendizaje profundo (perceptrón multicapa (MLP) y la memoria a largo plazo (LSTM).). Los modelos se implementaron con Python y utilizaron scikit-learn como paquete de aprendizaje automático y TensorFlow como paquete de aprendizaje profundo. Concluyen que "al observar las dos métricas de rendimiento del tiempo de entrenamiento y RMSE, se descubre que agregar más información sobre fechas y precios es útil para la predicción de ventas, pero los modelos de aprendizaje automático y aprendizaje profundo no tienen ventajas obvias en la predicción de ventas".

Yi (2023), presenta un modelo de predicción de ventas de Walmart, utilizando modelos de regresión lineal, regresión de bosques aleatorios y regresión de XGBoost, este último obteniendo los mejores resultados en rendimiento y precisión.

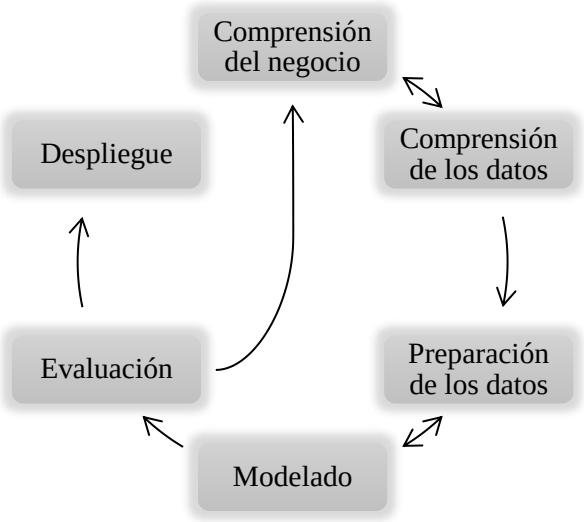
Sai Nikhil Boyapati & Ramesh Mummidi (2020), tienen como objetico identificar características críticas que influyen en las ventas, además de buscar un algoritmo adecuado para la previsión de ventas. Para ello, utilizaron la herramienta Python aplicaron algoritmos de aprendizaje automático supervisado como la regresión lineal simple obteniendo una precisión de 84%, la regresión con gradiente potenciando obtuvo una precisión de 91,2%, la regresión con vectores de soporte obtuvo una precisión de 88,4%, y la regresión de bosque aleatorio obtuvo una precisión de 91,5%, siendo este el más o para este caso.

Metodología

Este proyecto adopto la metodología CRISP-DM (Cross Industry Standard Process for Data Mining), ampliamente reconocida en el ámbito de la ciencia de datos por su enfoque estructurado y flexible. Este enfoque se articuló en seis fases secuenciales, adaptadas a las necesidades del estudio.

Figura 1

Metodología CRISP-DM



1. Comprensión del negocio: Se definió como objetivo general la construcción de modelos predictivos que permitan anticipar variables clave del proceso de venta en empresas de comercio electrónico, en particular sobre la base de datos pública del marketplace Amazon. Esta etapa permitió identificar las necesidades del negocio: optimizar el inventario, anticipar la demanda y ajustar estrategias comerciales con base en predicciones confiables.

2. Comprensión de los datos: Se exploró el conjunto de datos, el cual contiene información de venta de ropa y accesorios de Amazon del año 2022 a diferentes estados de la India. Esta base de datos fue descargada de un repositorio en Github llamado “Amazon Sales Analysis” (Leon Jose, 2023).

3. Preparación de los datos: Se realizaron tareas de limpieza, transformación, creación de nuevas variables y codificación de categorías. También se aplicó balanceo de clases donde fue necesario. Este proceso permitió generar subconjuntos de datos adecuados para clasificación y regresión.

4. Modelado: Se construyeron dos tipos de modelos:

- Modelos de clasificación: se utilizaron para predecir variables categóricas como la categoría y la talla, se implementaron los siguientes algoritmos: Regresión Logística, Árbol de Decisión, K-Nearest Neighbors (KNN), Random Forest Classifier. Estos modelos fueron entrenados utilizando un conjunto de datos preprocesado y evaluados mediante métricas de desempeño.
- Modelos de regresión: Para predecir variables numéricas como la cantidad promedio vendida (Qty) por producto o por talla, se utilizaron modelos de regresión como: Regresión Lineal, Árbol de Decisión Regressor, KNN Regressor, Random Forest Regressor. Estos modelos permiten estimar el comportamiento futuro de la demanda de productos y orientar decisiones estratégicas de inventario.

5. Evaluación: Los modelos fueron validados utilizando métricas apropiadas para cada tipo de tarea: accuracy, precision, recall, F1-score y validación cruzada para clasificación; R^2, MAE, MSE y RMSE para regresión. Se comparó el desempeño de los modelos para seleccionar el más robusto en cada caso.

6. Despliegue: Si bien el modelo no fue implementado en un sistema de producción, se entregó una propuesta técnica para su futura integración en plataformas de comercio electrónico, incluyendo posibles escenarios de implementación y monitoreo continuo.

Transformación de los Datos

La base de datos a trabajar contiene información de venta de ropa y accesorios de Amazon de los meses marzo, abril, mayo y junio del año 2022 a estados de la India, el archivo está en formato CSV, contiene 128.976 registros y 21 variables, que incluyen información como: ID de pedido, Fecha, Estado, Canal de ventas, Producto, Tamaño, Cantidad, Importe, Ciudad de envío, entre otras.

Figura 2

Base de Datos Antes de Transformación

index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	Qty	currency	Amount	ship-city	ship-state	ship-postal-code	ship-country	B2B	fulfilled-by	New	PendingS
0	405-8078784-5731545	4/30/2022	Cancelled	Merchant	Amazon.in	Standard	T-shirt	S	On the Way	0	INR	647.62	MUMBAI	MAHARASHTRA	400081	IN	FALSE	Easy Ship		
1	171-9198151-1101146	4/30/2022	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	3XL	Shipped	1	INR	406	BENGALURU	KARNATAKA	560085	IN	FALSE	Easy Ship		
2	404-0687676-7273146	4/30/2022	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped	1	INR	329	NAVI MUMBAI	MAHARASHTRA	410210	IN	TRUE			
3	403-9615377-8133951	4/30/2022	Cancelled	Merchant	Amazon.in	Standard	Blazzer	L	On the Way	0	INR	753.33	PUDUCHERRY	PUDUCHERRY	605008	IN	FALSE	Easy Ship		
4	407-1069790-7240320	4/30/2022	Shipped	Amazon	Amazon.in	Expedited	Trousers	3XL	Shipped	1	INR	574	CHENNAI	TAMIL NADU	600073	IN	FALSE			
5	404-1490984-4578765	4/30/2022	Shipped	Amazon	Amazon.in	Expedited	T-shirt	XL	Shipped	1	INR	824	GHAZIABAD	UTTAR PRADESH	201102	IN	FALSE			
6	408-5748499-6859555	4/30/2022	Shipped	Amazon	Amazon.in	Expedited	T-shirt	L	Shipped	1	INR	653	CHANDIGARH	CHANDIGARH	160036	IN	FALSE			
7	406-7807733-3785945	4/30/2022	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	S	Shipped	1	INR	399	HYDERABAD	TELANGANA	500032	IN	FALSE	Easy Ship		
8	407-5443024-5233168	4/30/2022	Cancelled	Amazon	Amazon.in	Expedited	T-shirt	3XL	Cancelled	0			HYDERABAD	TELANGANA	500008	IN	FALSE			
9	402-4393761-0311520	4/30/2022	Shipped	Amazon	Amazon.in	Expedited	Shirt	XXL	Shipped	1	INR	363	Chennai	TAMIL NADU	600041	IN	FALSE			
10	407-5633625-6970741	4/30/2022	Shipped	Amazon	Amazon.in	Expedited	Shirt	S	Shipped	1	INR	685	CHENNAI	TAMIL NADU	600073	IN	FALSE			
11	171-4638481-6326716	4/30/2022	Shipped	Amazon	Amazon.in	Expedited	Shirt	XS	Shipped	1	INR	364	NOIDA	UTTAR PRADESH	201303	IN	FALSE			
12	405-5513694-8146768	4/30/2022	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	XS	Shipped	1	INR	399	Amravati.	MAHARASHTRA	444606	IN	FALSE	Easy Ship		
13	408-7955685-3083534	4/30/2022	Shipped	Amazon	Amazon.in	Expedited	T-shirt	XS	Shipped	1	INR	657	MUMBAI	MAHARASHTRA	400053	IN	FALSE			
14	408-1298370-1920302	4/30/2022	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	T-shirt	L	Shipped	1	INR	771	MUMBAI	MAHARASHTRA	400053	IN	FALSE	Easy Ship		
15	403-4965581-9520319	4/30/2022	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	6XL	Shipped	1	INR	544	GUNTAKAL	ANDHRA PRADESH	515801	IN	FALSE	Easy Ship		
16	406-9379318-6555504	4/30/2022	Shipped	Amazon	Amazon.in	Expedited	Shirt	XXL	Shipped	1	INR	329	JAIPUR	RAJASTHAN	302020	IN	FALSE			
17	405-9013803-8009918	4/30/2022	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped	1	INR	399	NEW DELHI	DELHI	110074	IN	FALSE			
18	402-4030358-5835511	4/30/2022	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	XXL	Shipped	1	INR	458	Gurgaon	HARYANA	122004	IN	FALSE	Easy Ship		
19	405-5957858-1051546	4/30/2022	Shipped	Amazon	Amazon.in	Expedited	T-shirt	XS	Shipped	1	INR	886	BENGALURU	KARNATAKA	560017	IN	FALSE			
20	405-0607769-0716360	4/30/2022	Shipped	Amazon	Amazon.in	Expedited	Shirt	3XL	Shipped	1	INR	517	TIRUCHIRAPPALL	TAMIL NADU	620018	IN	FALSE			
21	404-8494550-5860325	4/30/2022	Shipped	Amazon	Amazon.in	Expedited	T-shirt	M	Shipped	1	INR	666	BENGALURU	KARNATAKA	560040	IN	FALSE			
22	171-1305077-2813934	4/30/2022	Shipped	Amazon	Amazon.in	Expedited	Shirt	L	Shipped	1	INR	376	HYDERABAD	TELANGANA	500072	IN	FALSE			

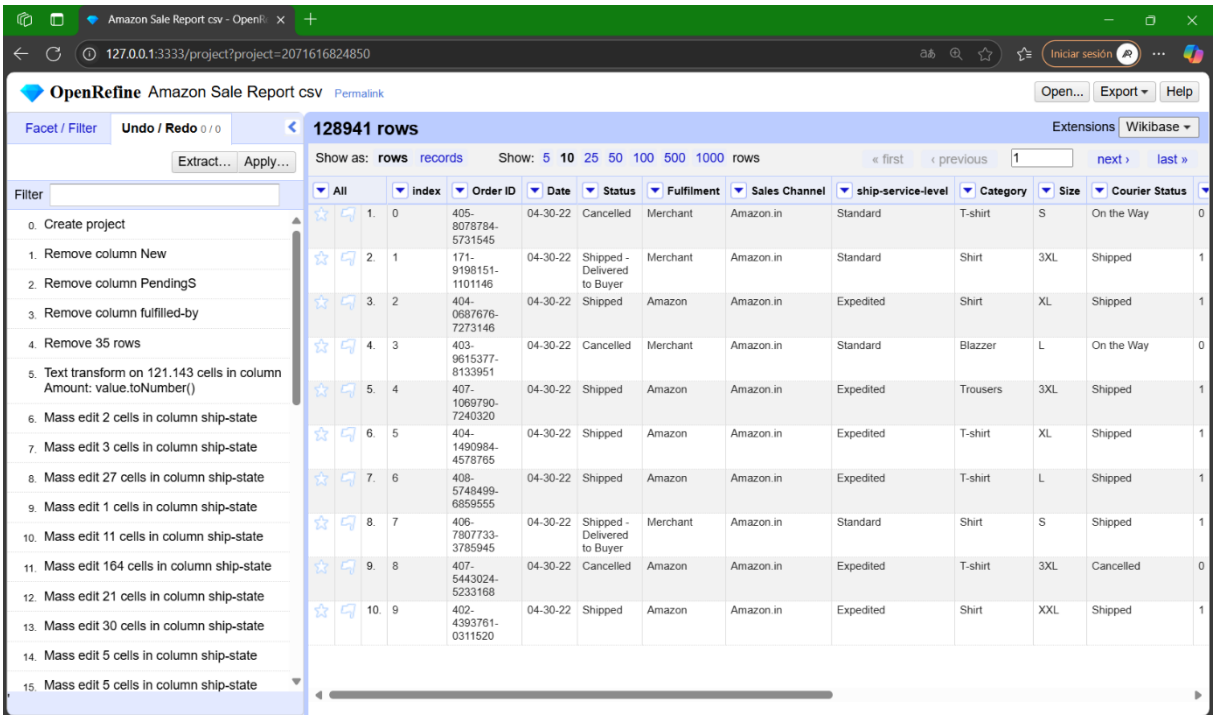
La base de datos fue cargada a la herramienta OpenRefine donde se realizaron los siguientes ajustes al archivo:

1. Se eliminaros las columnas “New” y “PendingS” dado que no contaban con ningún tipo de datos.
2. Se elimino la columna “fulfilled-by” dado que trata lo mismo de la columna “Fulfilment” pero con datos faltantes.
3. Se eliminan 35 filas vacías por no tener datos de la ciudad de destino.
4. Se transformo la columna “Amount” en formato número.
5. Agrupación de los países con el mismo nombre en la columna “ship-state”

6. Se descargo el archivo llamado “Amazon_Sale_Report_Limpio.csv”

Figura 3

Limpieza de Base de Datos con OpenRefine



Luego se cargó el archivo en formato CSV a Jupyter Notebook, con Python se realizaron los siguientes cambios:

1. Se cambio el formato de la columna “Date” en tipo fecha.

2. Se eliminaron filas repetidas de la columna “index”

3. Se toma la columna “B2B” la cual es de tipo booleano y se convierte a entero dando el valor de “1” cuando es “True” y “0” cuando es “False”.

4. Se descargo un nuevo archivo llamado “Amazon_Sale_Report_Limpio2.csv”

Figura 4

Limpieza de Base de Datos con Python

```
df = pd.read_csv('Amazon_Sale_Report_Limpio.csv', sep = ',', encoding = 'latin-1')
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
df = df.drop_duplicates(subset=['index'])
df['B2B'] = df['B2B'].astype(int)
df.head()
```

:x	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	Qty	currency	Amount	ship-city	ship-state	ship-postal-code	ship-country	B2B
0	405-8078784-5731545	2022-04-30	Cancelled	Merchant	Amazon.in	Standard	T-shirt	S	On the Way	0	INR	647.62	MUMBAI	MAHARASHTRA	400081	IN	0
1	171-9198151-1101146	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	3XL	Shipped	1	INR	406.00	BENGALURU	KARNATAKA	560085	IN	0
2	404-0687676-7273146	2022-04-30	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped	1	INR	329.00	NAVI MUMBAI	MAHARASHTRA	410210	IN	1
3	403-9615377-8133951	2022-04-30	Cancelled	Merchant	Amazon.in	Standard	Blazzer	L	On the Way	0	INR	753.33	PUDUCHERRY	PUDUCHERRY	605008	IN	0
4	407-1069790-7240320	2022-04-30	Shipped	Amazon	Amazon.in	Expedited	Trousers	3XL	Shipped	1	INR	574.00	CHENNAI	TAMIL NADU	600073	IN	0

Análisis Exploratorio de los Datos

Se realizó un análisis exploratorio de los datos (EDA) utilizando la herramienta Python y sus bibliotecas como Pandas, NumPy, Matplotlib y Seaborn, con el objetivo de comprender el comportamiento de las ventas registradas en la base de datos trabajada. Este análisis permitió identificar patrones de compra por categoría de producto y talla, preferencia de logística de envió, distribución de ventas por fechas y estados. Asimismo, se detectaron posibles valores atípicos y datos faltantes que fueron tratados para garantizar la calidad del conjunto de datos. La visualización de los datos facilitó la interpretación de las tendencias del mercado, proporcionando una base para la posterior construcción de modelos de clasificación y regresión enfocados en predecir comportamientos de compra y optimizar la gestión de inventario.

A continuación, se describen y muestran las diferentes visualizaciones generadas para la comprensión y análisis de los datos:

- Se realizo una tabla con la suma de las cantidades de cada talla para cada categoría.

A continuación, se describe la talla que más piden por cada categoría, de más a menos:

- Blazzer: L - M - XL - S - XXL - 3XL - XS
- Shirt: L - XL - M - XXL - S - 3XL - XS - 6XL - 5XL - 4XL
- Socks: L - S - XL - XXL - 3XL - M - XS
- T-shirt: M - L - XL - S - XXL - 3XL - XS - 6XL - 5XL - 4XL
- Trousers: XL - M - XXL - L - S - 3XL – XS
- Perfume, Shoes, Wallet y Watch: No aplica talla

Figura 5

Productos por Categoría y Talla

```
pivot_table = df.pivot_table(index='Category', columns='Size', values='Qty', aggfunc='sum', fill_value=0)
pivot_table
```

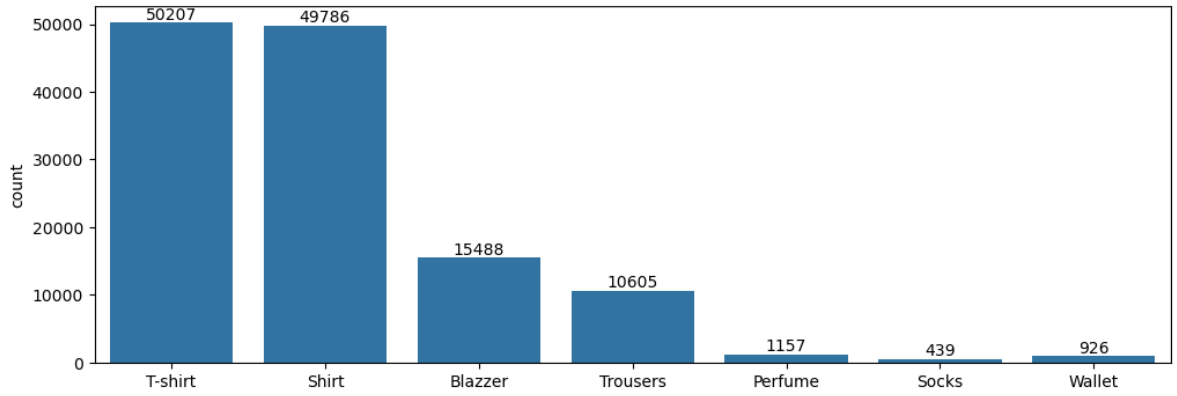
Size	3XL	4XL	5XL	6XL	Free	L	M	S	XL	XS	XXL
Category											
Blazzer	1758	0	0	0	0	2543	2324	1962	2123	1279	1944
Perfume	0	0	0	0	1051	0	0	0	0	0	0
Shirt	5125	350	456	616	0	8031	7699	5153	7825	2778	6931
Shoes	0	0	0	0	152	0	0	0	0	0	0
Socks	60	0	0	0	0	67	50	67	64	26	64
T-shirt	5282	46	57	72	0	7385	8346	6691	6863	4877	5598
Trousers	1120	0	0	0	0	1645	1692	1145	1731	868	1686
Wallet	0	0	0	0	863	0	0	0	0	0	0
Watch	0	0	0	0	3	0	0	0	0	0	0

- Se establece que productos son los más pedidos, obteniendo que la categoría que los clientes más solicitan es “T-shirt”, seguida de la “Shirt”, lo que menos solicitan es “Socks”.

Figura 6

Histograma de la Columna “Category”

```
Categoria_filtro = df[~df['Category'].isin(['Shoes', 'Watch'])]
plt.figure(figsize=(12, 4))
cat=sns.countplot(x='Category', data=Categoria_filtro)
for bars in cat.containers:
    cat.bar_label(bars)
```

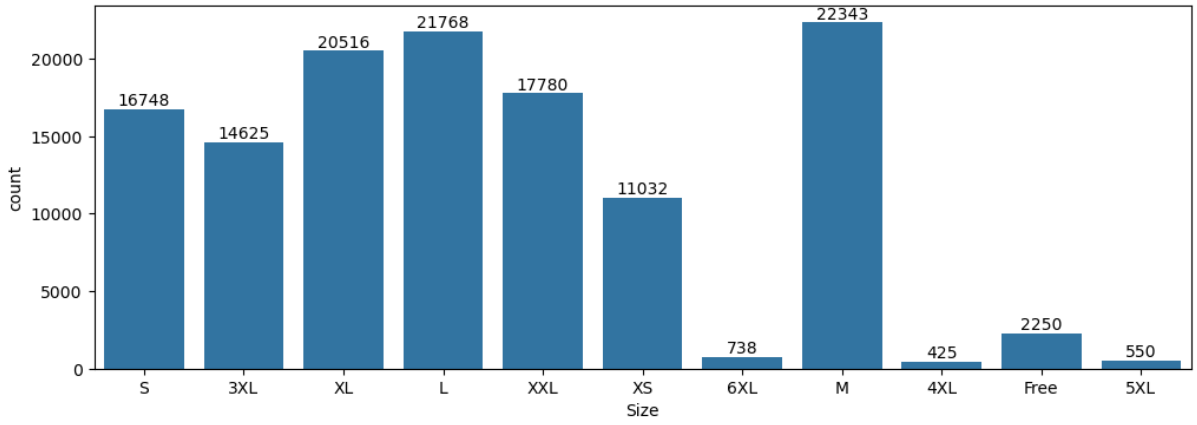


- Se establece que talla son las más pedidas, obteniendo que la talla que los clientes más solicitan es la “M”, seguida de la “L” y “XL”, la que menos solicitan es la “4XL”.

Figura 7

Histograma de la Columna “Size”

```
plt.figure(figsize=(12, 4))
cat=sns.countplot(x='Size', data=df)
for bars in cat.containers:
    cat.bar_label(bars)
```



- Cantidad de pedidos por mes y rango fecha de la base de datos a analizar: Los datos que se están trabajando van desde el 31 de marzo del 2022 al 29 de junio de 2022, el mes donde más pedidos hubo fue en abril.

Figura 8

Cantidad de Pedidos por Fecha

```
print("\nCantidad pedidos por mes:")
print(df['Date'].dt.to_period('M').value_counts().sort_index())
print("\nFecha inicio:", df['Date'].min())
print("Fecha fin:", df['Date'].max())
```

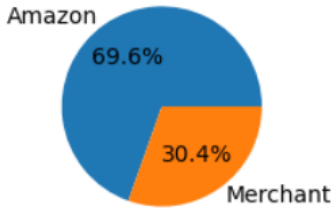
```
Cantidad pedidos por mes:
Date
2022-03      171
2022-04    48887
2022-05    42028
2022-06    37689
Freq: M, Name: count, dtype: int64

Fecha inicio: 2022-03-31 00:00:00
Fecha fin: 2022-06-29 00:00:00
```

- Se identificó la preferencia de compra si es directamente a Amazon o a un Merchant (vendedor): Los clientes prefieren en mayor medida comprarle directamente de Amazon y que ellos se encarguen de la logística de envío.

Figura 9

Diagrama de la Columna "Fulfilment"

```
Fulfilment = df['Fulfilment'].value_counts()
plt.figure(figsize=(2, 2))
plt.pie(Fulfilment, labels=Fulfilment.index, autopct='%1.1f%%')
plt.show()
```

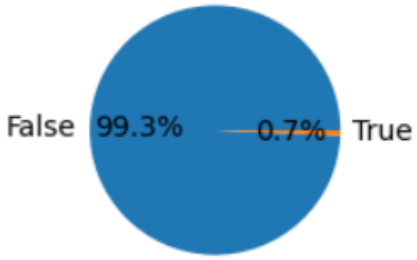


- Ventas a empresas o consumidor final: El 99% de las ventas son directamente al consumidor final.

Figura 10

Diagrama de la Columna "B2B"

```
B2B = df['B2B'].value_counts()
plt.figure(figsize=(2, 2))
plt.pie(B2B, labels=B2B.index, autopct='%1.1f%%')
plt.show()
```

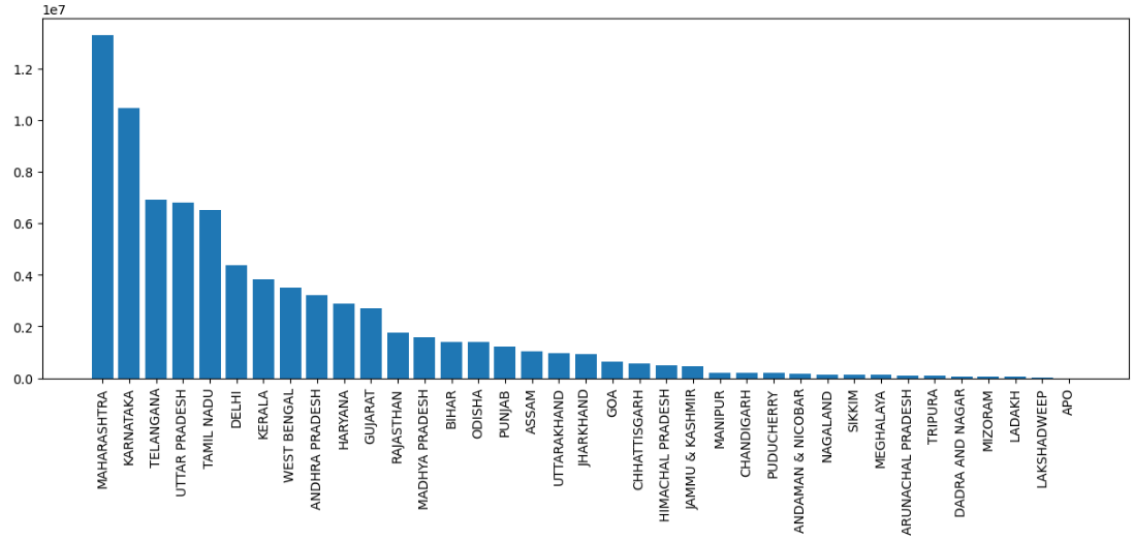


- Estado de la India con mayor compra es Maharashtra, seguido del estado Karnataka.

Figura 11

Histograma de la Columna "Ship-State"

```
ventas_por_estado = df.groupby('ship-state')['Amount'].sum().sort_values(ascending=False).reset_index()
plt.figure(figsize=(15, 5))
plt.bar(ventas_por_estado['ship-state'], ventas_por_estado['Amount'])
plt.xticks(rotation=90)
plt.show()
```



- La cantidad mínima pedida por producto es 1 y la cantidad máxima es 15, sin embargo, en promedio los clientes solo piden una unidad por producto.

Figura 12

Estadísticas de la Columna "Qty"

```
Qty_filtro = df[df['Qty'] != 0]
metricas_numericas = Qty_filtro['Qty'].describe(percentiles=[.25, .5, .75])
print(metricas_numericas)
```

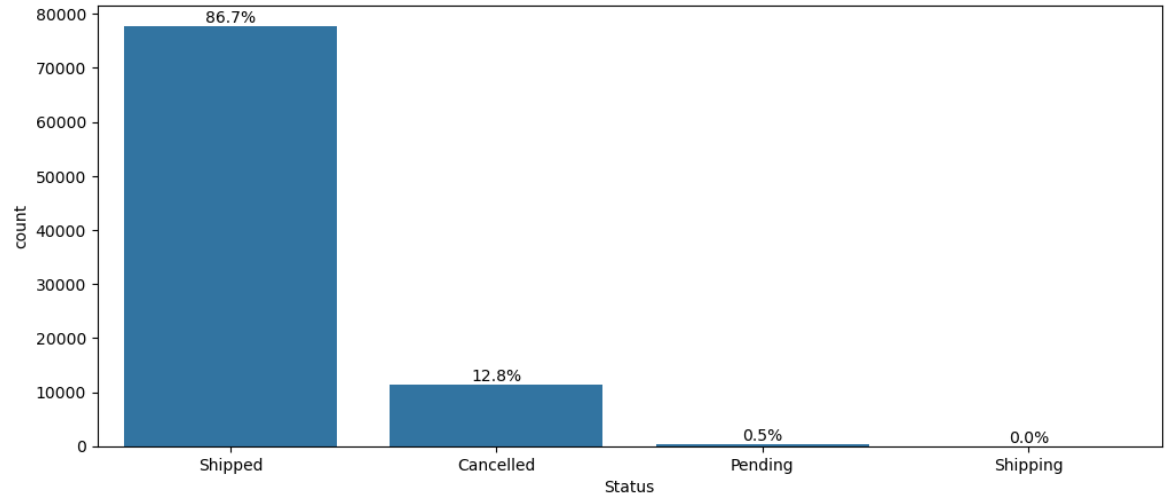
```
count    115988.000000
mean          1.004138
std           0.094327
min           1.000000
25%           1.000000
50%           1.000000
75%           1.000000
max          15.000000
Name: Qty, dtype: float64
```

- Estado de los pedidos solicitados directamente a Amazon: El 86.7% de los pedidos ya fueron enviados y un 12.8% de los pedidos fue cancelado.

Figura 13

Estado de los Pedidos por "Fulfilment = Amazon"

```
df_Amazon = df[df['Fulfilment'] == 'Amazon']
total = len(df_Amazon)
plt.figure(figsize=(12, 5))
AM = sns.countplot(data=df_Amazon, x='Status')
for bars in AM.containers:
    for bar in bars:
        height = bar.get_height()
        percentage = f'{(height / total) * 100:.1f}%'
        AM.text(bar.get_x() + bar.get_width() / 2, height, percentage,
                ha='center', va='bottom', fontsize=10)
plt.show()
```

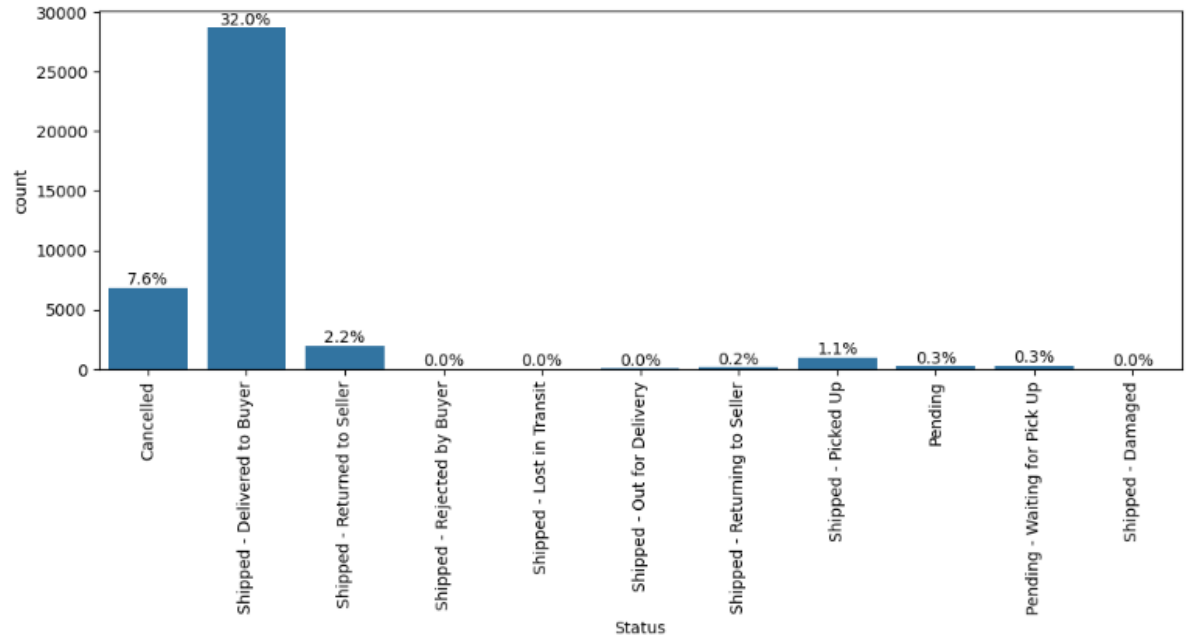


- Estado de los pedidos solicitados directamente a un vendedor: El 32% de los pedidos fueron entregados al comprador, el 7.6% de los pedidos fueron cancelados, el 2.4% de los pedidos fueron devueltos al vendedor o está en proceso de devolución.

Figura 14

Estado de los Pedido por "Fulfilment = Merchant"

```
df_Merchant = df[df['Fulfilment'] == 'Merchant']
vtotal = len(df_Merchant)
plt.figure(figsize=(12, 4))
ME = sns.countplot(data=df_Merchant, x='Status')
for bars in ME.containers:
    for bar in bars:
        height = bar.get_height()
        percentage = f'{(height / total) * 100:.1f}%'
        ME.text(bar.get_x() + bar.get_width() / 2, height, percentage,
                ha='center', va='bottom', fontsize=10)
plt.xticks(rotation=90)
plt.show()
```



- Promedio de ventas mensual por categoría y talla agrupada, se puede observar que la categoría Shirt en talla mediana fue la que más solicitaron, seguido de T-shirt pequeñas y se vendió en menor cantidad Socks grandes.

Figura 15

Promedio de Cantidad Vendida Mensual por Categoría y Talla

```
df0 = df.copy()
df0 = df0.dropna()
df0 = df0[df0['Qty'] > 0]
df0 = df0[~df0['Category'].isin(['Shoes', 'Watch'])]

def agrupar_talla(talla):
    if talla in ['XS', 'S', 'M']:
        return 'Pequeña'
    elif talla in ['L', 'XL', 'XXL']:
        return 'Mediana'
    elif talla in ['Free']:
        return 'NA'
    else:   # 3XL, 4XL, 5XL, 6XL
        return 'Grande'
df0['Grupo_Talla'] = df0['Size'].apply(agrupar_talla)

df0['Date'] = pd.to_datetime(df0['Date'], errors='coerce')
df0['Mes'] = df0['Date'].dt.month
df0 = df0[df0['Mes'].isin([4, 5, 6])]

ventas_mensuales = df0.groupby(['Category', 'Grupo_Talla', 'Mes'])['Qty'].sum().reset_index()
promedios = ventas_mensuales.groupby(['Category', 'Grupo_Talla'])['Qty'].mean().reset_index()
promedios.rename(columns={'Qty': 'Promedio_Mes_Qty'}, inplace=True)
promedios['Promedio_Mes_Qty'] = promedios['Promedio_Mes_Qty'].round(2)
promedios
```

	Category	Grupo_Talla	Promedio_Mes_Qty
0	Blazzer	Grande	585.33
1	Blazzer	Mediana	2202.33
2	Blazzer	Pequeña	1853.33
3	Perfume	NA	350.00
4	Shirt	Grande	2176.00
5	Shirt	Mediana	7564.67
6	Shirt	Pequeña	5198.67
7	Socks	Grande	20.00
8	Socks	Mediana	65.00
9	Socks	Pequeña	47.33
10	T-shirt	Grande	1815.00
11	T-shirt	Mediana	6596.67
12	T-shirt	Pequeña	6616.67
13	Trousers	Grande	371.67
14	Trousers	Mediana	1686.00
15	Trousers	Pequeña	1233.67
16	Wallet	NA	281.00

Desarrollo y Evaluación de los Modelos de Aprendizaje Automático

En esta sección se describe la implementación y entrenamiento de diferentes modelos de aprendizaje automático utilizando Python, con el objetivo de predecir aspectos clave relacionados con las ventas de productos. Se utilizaron tanto algoritmos de clasificación como de regresión, dependiendo del tipo de variable objetivo. Los modelos fueron ajustados y evaluados mediante distintas métricas de desempeño.

Para cada uno de los modelos implementados, se realizó un proceso de ajuste de hiperparámetros y validación cruzada para mejorar su rendimiento y evitar sobreajuste. Finalmente, se compararon los modelos entre sí con base en sus métricas, seleccionando los que ofrecieron mejor desempeño en términos de estabilidad y precisión.

Modelos de Clasificación

Las variables utilizadas fueron:

- ship-state: El estado de la India a donde se envió el producto.
- ship-service-level: El nivel de servicio de envío.
- Fulfilment: Información sobre quien es el encargado de la logística de envió.
- Qty: La cantidad de unidades por pedido.
- Amount: El monto total de la venta.
- B2B: Un indicador booleano de si la venta fue a una empresa o al consumidor final.
- Category: que representa los artículos de ropa (T-shirt, Shirt, Blazzer, Trousers, Socks).
- Size: talla del producto (esta variable fue codificada utilizando one-hot encoding para convertirla en formato numérico y que el modelo pudiera procesar).

Las variables Category y Size se convierten en variables de salida, según lo que se requiere predecir.

En todos los modelos el conjunto de datos se dividió en un 80% para entrenamiento y un 20% para pruebas. Se estableció una semilla para la aleatoriedad (random_state=45) asegurando que la división sea la misma cada vez que ejecutes el código, y se imprime un resumen del rendimiento del modelo en el conjunto de prueba

También se aplicó la validación cruzada, el cual evalúa el rendimiento del modelo en diferentes subconjuntos, en este caso hacen 5 divisiones y se imprime el promedio de la precisión.

Predicción de la Columna "Category"

El objetivo es predecir qué productos se van a vender en mayor medida, basándose en diversas características de ventas, lo cual es útil para la gestión de inventario, dado que al saber qué tipo de producto es probable que se venda, se puede asegurar la tenencia del stock adecuado, organizar eficientemente las bodegas y anticipar las necesidades de reabastecimiento.

Modelado:

- Regresión logística

En este modelo se crea un pipeline, primero se estandarizan los datos numéricos para que todas las características tengan la misma escala, luego se aplica el algoritmo de clasificación, se hace un balance para ayuda a manejar desequilibrios si algunas categorías tienen muchos más ejemplos que otras, y se da un máximo de 2000 iteraciones para que el modelo tenga más oportunidades de converger.

Figura 16

Regresión Logística para Predicción de "Category"

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split, cross_validate
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn.metrics import classification_report

df_ml = df2.copy()

y_ml = df_ml['Category']

x_ml = df_ml[['Size', 'ship-state', 'ship-service-level', 'Fulfilment', 'Qty', 'Amount', 'B2B']]
x_ml = pd.get_dummies(x_ml, columns=['Size'], drop_first=True)

x_ml_train, x_ml_test, y_ml_train, y_ml_test = train_test_split(x_ml, y_ml, test_size=0.2, random_state=45)

mod_ml = make_pipeline(StandardScaler(), LogisticRegression(max_iter=2000, class_weight='balanced', random_state=45))

scores_ml = cross_validate(mod_ml, x_ml, y_ml, cv=5, scoring=['accuracy'])
print("Accuracy promedio:", scores_ml['test_accuracy'].mean())

mod_ml.fit(x_ml_train, y_ml_train)
y_ml_pred = mod_ml.predict(x_ml_test)
print("\nReporte de clasificación:\n", classification_report(y_ml_test, y_ml_pred, zero_division=0))
```

```
Accuracy promedio: 0.44027145576429694

Reporte de clasificación:
              precision    recall  f1-score   support

     Blazzer       0.33      0.43      0.37      2944
     Perfume       0.78      0.68      0.72       226
       Shirt       0.85      0.35      0.50      9342
       Socks       0.01      0.73      0.03        75
     T-shirt       0.71      0.53      0.61      9372
    Trousers       0.18      0.47      0.27      2029
      Wallet       0.65      0.75      0.70       178

    accuracy                           0.45     24166
   macro avg       0.50      0.56      0.46     24166
weighted avg       0.67      0.45      0.51     24166
```

- Árbol de decisión

En este modelo se crea un pipeline, primero se estandarizan los datos numéricos, luego un árbol de decisión toma decisiones secuenciales basadas en las características para clasificar la categoría, se limita la profundidad del árbol a 6 niveles y se hace un balance.

Figura 17

Árbol de Decisión para Predicción de "Category"

```
from sklearn.model_selection import train_test_split, cross_validate
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
from sklearn.pipeline import make_pipeline

df_ad = df2.copy()

y_ad = df_ad['Category']

x_ad = df_ad[['Size', 'ship-state', 'ship-service-level', 'Fulfilment', 'Qty', 'Amount', 'B2B']]
x_ad = pd.get_dummies(x_ad, columns=['Size'], drop_first=True)

x_ad_train, x_ad_test, y_ad_train, y_ad_test = train_test_split(x_ad, y_ad, test_size=0.2, random_state=45)

mod_ad = make_pipeline(StandardScaler(), DecisionTreeClassifier(max_depth=6, class_weight='balanced', random_state=45))

scores_ad = cross_validate(mod_ad, x_ad, y_ad, cv=5, scoring=['accuracy'])
print("Accuracy promedio:", scores_ad['test_accuracy'].mean())

mod_ad.fit(x_ad_train, y_ad_train)
y_ad_pred = mod_ad.predict(x_ad_test)
print("\nReporte de clasificación:\n", classification_report(y_ad_test, y_ad_pred, zero_division=0))
```

```
Accuracy promedio: 0.5741454936687909

Reporte de clasificación:
              precision    recall  f1-score   support

     Blazzer       0.42      0.75      0.54      2944
     Perfume       0.89      0.96      0.92       226
       Shirt       0.86      0.49      0.62      9342
       Socks       0.03      0.92      0.05        75
     T-shirt       0.80      0.60      0.69      9372
    Trousers       0.26      0.45      0.33      2029
      Wallet       0.95      0.84      0.89       178

    accuracy                           0.57     24166
   macro avg       0.60      0.72      0.58     24166
weighted avg       0.73      0.57      0.62     24166
```

- Random Forest

En este modelo se crea un pipeline, primero se estandarizan los datos numéricos, luego un Random Forest construye múltiples árboles de decisión y combina sus predicciones para obtener un resultado más preciso y robusto, se configuraron 200 árboles de decisión para construir el bosque y se hace un balance.

Figura 18

Random Forest para Predicción de "Category"

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from sklearn.model_selection import cross_validate
from sklearn.pipeline import make_pipeline

df_rf = df2.copy()

y_rf = df_rf['Category']

x_rf = df_rf[['Size', 'ship-state', 'ship-service-level', 'Fulfilment', 'Qty', 'Amount', 'B2B']]
x_rf = pd.get_dummies(x_rf, columns=['Size'], drop_first=True)

x_rf_train, x_rf_test, y_rf_train, y_rf_test = train_test_split(x_rf, y_rf, test_size=0.2, random_state=45)

mod_rf = make_pipeline(StandardScaler(),
                       RandomForestClassifier(n_estimators=200, class_weight='balanced', random_state=45))

scores_rf = cross_validate(mod_rf, x_rf, y_rf, cv=5, scoring=['accuracy'])
print("Accuracy promedio:", scores_rf['test_accuracy'].mean())

mod_rf.fit(x_rf_train, y_rf_train)
y_rf_pred = mod_rf.predict(x_rf_test)
print("\nReporte de clasificación:\n", classification_report(y_rf_test, y_rf_pred))
```

```
Accuracy promedio: 0.7479516676322107

Reporte de clasificación:
              precision    recall  f1-score   support

     Blazzer       0.71      0.76      0.73      2944
     Perfume       0.92      0.95      0.94       226
       Shirt       0.85      0.80      0.82      9342
       Socks       0.06      0.21      0.09        75
     T-shirt       0.84      0.82      0.83      9372
    Trousers       0.50      0.58      0.54      2029
      Wallet       0.94      0.90      0.92       178

    accuracy                           0.78     24166
   macro avg       0.69      0.72      0.70     24166
weighted avg       0.80      0.78      0.79     24166
```

- KNN

En este modelo se crea un pipeline, primero se estandarizan los datos numéricos, luego se aplica el algoritmo KNN y se establece k=5, lo que significa que el modelo considero los 5 puntos de datos más cercanos en el conjunto de entrenamiento para determinar la categoría.

Figura 19

KNN para Predicción de "Category"

```
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn.metrics import classification_report
from sklearn.model_selection import cross_validate

df_knn = df2.copy()

y_knn = df_knn['Category']

x_knn = df_knn[['Size', 'ship-state', 'ship-service-level', 'Fulfilment', 'Qty', 'Amount', 'B2B']]
x_knn = pd.get_dummies(x_knn, columns=['Size'], drop_first=True)

x_knn_train, x_knn_test, y_knn_train, y_knn_test = train_test_split(x_knn, y_knn, test_size = 0.2, random_state = 45)

mod_knn = make_pipeline(StandardScaler(), KNeighborsClassifier(n_neighbors=5))

scores_knn = cross_validate(mod_knn, x_knn, y_knn, cv=5, scoring=['accuracy'])
print("Accuracy promedio:", scores_knn['test_accuracy'].mean())

mod_knn.fit(x_knn_train, y_knn_train)
y_knn_pred = mod_knn.predict(x_knn_test)
print("\nReporte de clasificación:\n", classification_report(y_knn_test, y_knn_pred))
```

```
Accuracy promedio: 0.7239013490027311

Reporte de clasificación:
              precision    recall  f1-score   support

     Blazzer       0.62      0.64      0.63      2944
     Perfume       0.86      0.82      0.84       226
       Shirt       0.79      0.85      0.82      9342
       Socks       0.00      0.00      0.00        75
     T-shirt       0.79      0.79      0.79      9372
    Trousers       0.54      0.31      0.39      2029
      Wallet       0.78      0.83      0.80       178

    accuracy                           0.75     24166
   macro avg       0.62      0.61      0.61     24166
weighted avg       0.74      0.75      0.75     24166
```

Evaluación: Una vez entrenados y obtenidas las métricas de desempeño de cada modelo, se procedió a realizar una comparación de su rendimiento para identificar el enfoque más adecuado para la predicción de categorías. La tabla a continuación resume las métricas clave de cada modelo:

Tabla 1

Predicción de la Columna "Category"

Modelo	Accuracy	Macro F1	Weighted F1	Validación cruzada	Observaciones
Logístico	0.45	0.46	0.51	0.44	Bajo rendimiento general, especialmente en clases desbalanceadas como Socks.
Árbol decisión	0.57	0.58	0.62	0.57	Mejora frente al logístico con buen desempeño en Blazzer, Perfume y Wallet.
Random Forest	0.78	0.70	0.79	0.75	Mejor desempeño general, equilibrio entre clases, robusto ante el desbalance.
K-NN	0.75	0.61	0.75	0.72	Buen desempeño, pero sensible al desbalance y dependiente del valor de K.

Tras evaluar los cuatro modelos de clasificación con el objetivo de predecir la variable "Category", se observaron diferencias significativas en el rendimiento de cada uno:

El modelo de Regresión Logística presentó un rendimiento considerablemente bajo en todas las métricas, con un accuracy de 0.45 y un macro F1 de 0.46, lo que evidencia una baja capacidad de generalización y una alta sensibilidad al desbalance de clases. Este modelo no resulta adecuado para este problema de clasificación multiclase.

El modelo de Árbol de Decisión mostró una mejora significativa respecto al logístico, alcanzando un accuracy de 0.57 y un weighted F1 de 0.62. Si bien se observaron avances en la clasificación de categorías como Blazzer y T-shirt, aún persisten limitaciones en el equilibrio general del modelo, especialmente con clases minoritarias.

El modelo K-Nearest Neighbors (K-NN) alcanzó un accuracy de 0.75 y un macro F1 de 0.61, demostrando un buen rendimiento general. No obstante, su desempeño fue más sensible al desbalance de clases en comparación con Random Forest. Aun así, se considera una alternativa válida, especialmente en escenarios donde se prioriza la simplicidad del modelo y su interpretabilidad.

Finalmente, el modelo de Random Forest fue el que presentó el mejor desempeño global, con un accuracy de 0.78, un weighted F1 de 0.79 y una puntuación de validación cruzada de 0.75, lo que sugiere una alta capacidad de generalización. Aunque las clases con menor representación, como Socks, fueron desafiantes para todos los modelos, Random Forest logró una mejor cobertura y equilibrio, consolidándose como el modelo más robusto y confiable para este conjunto de datos.

Predicción de la Columna "Size_Agrupada"

El objetivo es predecir qué tallas se van a vender en mayor medida. Se implantaron los modelos anteriores, sin embargo, al haber 10 tipos de tallas y desbalance entre ellas, los modelos no funcionaron bien, sesgándose hacia las tallas con más datos, por ello, se procedió a agrupar las tallas pasando de 10 tallas a 3 grupos (pequeña, mediana y grande), buscando simplificar el análisis de la distribución de las tallas y facilitar la identificación de patrones de demanda a un nivel más general, la nueva columna se denominó "Grupo_Talla". No se tuvieron en cuenta los productos que no les aplica talla.

Figura 20

Agrupación de "Size"

```
df3 = pd.read_csv('Amazon_Sale_Report_Limpio2.csv', sep = ',', encoding = 'latin-1')
df3 = df3.dropna()
df3 = df3[df3['Size'] != 'Free']
df3["Fulfilment"] = df3["Fulfilment"].map({"Merchant": 0, "Amazon": 1})
df3["ship-service-level"] = df3["ship-service-level"].map({"Standard": 0, "Expedited": 1})
df3["ship-state"] = df3["ship-state"].astype("category").cat.codes
```

```
def agrupar_talla(talla):
    if talla in ['XS', 'S', 'M']:
        return 'Pequeña'
    elif talla in ['L', 'XL', 'XXL']:
        return 'Mediana'
    else:   # 3XL, 4XL, 5XL, 6XL
        return 'Grande'

df3['Grupo_Talla'] = df3['Size'].apply(agrupar_talla)
```

```
df3.head()
```

Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	Qty	currency	Amount	ship-city	ship-state	ship-postal-code	ship-country	B2B	Grupo_Talla
405-8078784-5731545	2022-04-30	Cancelled	0	Amazon.in	0	T-shirt	S	On the Way	0	INR	647.62	MUMBAI	20	400081	IN	0	Pequeña
171-9198151-1101146	2022-04-30	Shipped - Delivered to Buyer	0	Amazon.in	0	Shirt	3XL	Shipped	1	INR	406.00	BENGALURU	15	560085	IN	0	Grande
404-0687676-7273146	2022-04-30	Shipped	1	Amazon.in	1	Shirt	XL	Shipped	1	INR	329.00	NAVI MUMBAI	20	410210	IN	1	Mediana

Modelado:

- Regresión logística

Figura 21

Regresión Logística para Predicción de "Size_Agrupada"

```
Accuracy promedio: 0.36768228449461626

Reporte de clasificación:
              precision    recall  f1-score   support

      Grande       0.16      0.39      0.22      3061
     Mediana       0.51      0.32      0.39     11406
     Pequeña       0.44      0.44      0.44      9305

    accuracy                           0.37     23772
   macro avg       0.37      0.38      0.35     23772
weighted avg       0.44      0.37      0.39     23772
```

- Árbol de decisión

Figura 22

Árbol de Decisión para Predicción de "Size_Agrupada"

```
Accuracy promedio: 0.39739865604314567

Reporte de clasificación:
              precision    recall  f1-score   support

      Grande       0.18      0.33      0.24      3061
     Mediana       0.52      0.39      0.45     11406
     Pequeña       0.44      0.47      0.45      9305

    accuracy                           0.41     23772
   macro avg       0.38      0.40      0.38     23772
weighted avg       0.45      0.41      0.42     23772
```

- Random Forest

Figura 23

Random Forest para Predicción de "Size_Agrupada"

```
Accuracy promedio: 0.40703200463862893

Reporte de clasificación:
              precision    recall  f1-score   support

      Grande       0.21      0.38      0.27      3061
     Mediana       0.54      0.44      0.49     11406
     Pequeña       0.46      0.44      0.45      9305

    accuracy                           0.43     23772
   macro avg       0.40      0.42      0.40     23772
weighted avg       0.47      0.43      0.44     23772
```

- KNN

Figura 24

KNN para Predicción de "Size_Agrupada"

```
Accuracy promedio: 0.43799322568934207

Reporte de clasificación:
              precision    recall  f1-score   support

      Grande       0.24      0.19      0.22      3061
     Mediana       0.51      0.60      0.55     11406
     Pequeña       0.45      0.38      0.41      9305

    accuracy                           0.46     23772
   macro avg       0.40      0.39      0.39     23772
weighted avg       0.45      0.46      0.45     23772
```

Evaluación: Una vez entrenados y obtenidas las métricas de desempeño de cada modelo, se procedió a realizar una comparación de su rendimiento para identificar el enfoque más adecuado para la predicción del grupo de talla. La tabla a continuación resume las métricas clave de cada modelo:

Tabla 2

Predicción de la Columna "Size_Agrupada"

Modelo	Accuracy	Macro F1	Weighted F1	Validación cruzada	Observaciones
Logístico	0.37	0.35	0.39	0.368	Bajo rendimiento, especialmente flojo en Grande.
Árbol decisión	0.41	0.38	0.42	0.397	Ligera mejora, pero aún pobre balance entre clases.

Modelo	Accuracy	Macro F1	Weighted F1	Validación cruzada	Observaciones
Random Forest	0.43	0.40	0.44	0.407	Mejor balance general, aunque aún con baja precisión.
K-NN	0.46	0.39	0.45	0.438	Mejor resultado en accuracy y f1, pero bajo en clase Grande.

Se evaluaron cuatro modelos de clasificación con el objetivo de predecir la variable "Size_Agrupada". Todos los modelos presentaron un rendimiento moderado a bajo con accuracy entre 37% y 46%.

Regresión Logística presentó el rendimiento más bajo (Accuracy de 0.37, Macro F1 de 0.35), con un desempeño particularmente deficiente en la clase Grande. Esto sugiere que el modelo no logra capturar adecuadamente las diferencias entre las clases.

Árbol de Decisión mostró una ligera mejora (Accuracy de 0.41), aunque el Macro F1 de 0.38 indica que persisten problemas de balance entre clases. Si bien mejora marginalmente la precisión, no logra una clasificación robusta.

Random Forest obtuvo un rendimiento algo superior (Accuracy de 0.43, Macro F1 de 0.40), destacándose por un mejor equilibrio general entre clases. Sin embargo, su precisión sigue siendo limitada, lo que sugiere que el modelo no logra capturar completamente la complejidad de la variable objetivo.

El modelo K-NN fue el mejor clasificador con un accuracy de 46% y un F1 weighted de 0.45, seguido de cerca por Random Forest. La clase Mediana, al ser la más frecuente, fue la mejor clasificada en todos los modelos. Por el contrario, la clase Grande mostró un bajo

desempeño en precisión y recall. A pesar de ello, se posiciona como la mejor alternativa entre los modelos evaluados.

Modelo de Regresión

Predicción de la Columna "Qty_Mensual"

Para predecir la cantidad mensual que se venderían por producto y talla, se tuvo en cuenta el dataframe llamado "ventas_mensuales" que se creó para el análisis exploratorio, las variables de entrada utilizadas fueron Category, Grupo_Talla y Mes, la variable de salida fue Qty_mensual, que representa la suma mensual de los artículos pedidos.

Modelado:

- Regresión lineal

Figura 25

Regresión Lineal para Predicción de "Qty_Mensual"

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error

y_ml = ventas_mensuales['Qty']

x_ml = ventas_mensuales[['Category', 'Grupo_Talla', 'Mes']]
x_ml = pd.get_dummies(x_ml, columns=['Category', 'Grupo_Talla'], drop_first=True)

x_ml_train, x_ml_test, y_ml_train, y_ml_test = train_test_split(x_ml, y_ml, test_size=0.2, random_state=45)

mod_ml = LinearRegression()
mod_ml.fit(x_ml_train, y_ml_train)
y_pred_ml = mod_ml.predict(x_ml_test)

R_cuadrado = r2_score(y_ml_test, y_pred_ml)
MAE = mean_absolute_error(y_ml_test, y_pred_ml)
MSE = mean_squared_error(y_ml_test, y_pred_ml)
RMSE = np.sqrt(mean_squared_error(y_ml_test, y_pred_ml))

print(f"R_cuadrado: {R_cuadrado:.4f}")
print(f"MAE: {MAE:.4f}")
print(f"MSE: {MSE:.4f}")
print(f"RMSE: {RMSE:.4f}")
```

```
R_cuadrado: 0.9168
MAE: 527.8905
MSE: 410697.0884
RMSE: 640.8565
```

- Árbol de decisión

Figura 26

Árbol de Decisión para Predicción de "Qty_Mensual"

```
from sklearn.tree import DecisionTreeRegressor

y_ad = ventas_mensuales['Qty']

x_ad = ventas_mensuales[['Category', 'Grupo_Talla', 'Mes']]
x_ad = pd.get_dummies(x_ad, columns=['Category', 'Grupo_Talla'], drop_first=True)

x_ad_train, x_ad_test, y_ad_train, y_ad_test = train_test_split(x_ad, y_ad, test_size=0.2, random_state=45)

mod_ad = DecisionTreeRegressor(random_state=42)
mod_ad.fit(x_ad_train, y_ad_train)
y_pred_ad = mod_ad.predict(x_ad_test)

R_cuadrado = r2_score(y_ad_test, y_pred_ad)
MAE = mean_absolute_error(y_ad_test, y_pred_ad)
MSE = mean_squared_error(y_ad_test, y_pred_ad)
RMSE = np.sqrt(mean_squared_error(y_ad_test, y_pred_ad))

print(f"R_cuadrado: {R_cuadrado:.4f}")
print(f"MAE: {MAE:.4f}")
print(f"MSE: {MSE:.4f}")
print(f"RMSE: {RMSE:.4f}")
```

```
R_cuadrado: 0.9054
MAE: 485.3636
MSE: 467348.8182
RMSE: 683.6292
```

- Random Forest

Figura 27

Random Forest para Predicción de "Qty_Mensual"

```
from sklearn.ensemble import RandomForestRegressor

y_rf = ventas_mensuales['Qty']

x_rf = ventas_mensuales[['Category', 'Grupo_Talla', 'Mes']]
x_rf = pd.get_dummies(x_rf, columns=['Category', 'Grupo_Talla'], drop_first=True)

x_rf_train, x_rf_test, y_rf_train, y_rf_test = train_test_split(x_rf, y_rf, test_size=0.2, random_state=45)

mod_rf = RandomForestRegressor(random_state=42)
mod_rf.fit(x_rf_train, y_rf_train)
y_pred_rf= mod_rf.predict(x_rf_test)

R_cuadrado = r2_score(y_rf_test, y_pred_rf)
MAE = mean_absolute_error(y_rf_test, y_pred_rf)
MSE = mean_squared_error(y_rf_test, y_pred_rf)
RMSE = np.sqrt(mean_squared_error(y_rf_test, y_pred_rf))

print(f"R_cuadrado: {R_cuadrado:.4f}")
print(f"MAE: {MAE:.4f}")
print(f"MSE: {MSE:.4f}")
print(f"RMSE: {RMSE:.4f}")
```

```
R_cuadrado: 0.8699
MAE: 492.9136
MSE: 642238.4547
RMSE: 801.3978
```

- KNN

Figura 28

KNN para Predicción de "Qty_Mensual"

```
from sklearn.neighbors import KNeighborsRegressor

y_knn = ventas_mensuales['Qty']

x_knn = ventas_mensuales[['Category', 'Grupo_Talla', 'Mes']]
x_knn = pd.get_dummies(x_knn, columns=['Category', 'Grupo_Talla'], drop_first=True)

x_knn_train, x_knn_test, y_knn_train, y_knn_test = train_test_split(x_knn, y_knn, test_size = 0.2, random_state = 45)

mod_knn = KNeighborsRegressor(n_neighbors=5)
mod_knn.fit(x_knn_train, y_knn_train)
y_pred_knn= mod_knn.predict(x_knn_test)

R_cuadrado = r2_score(y_knn_test, y_pred_knn)
MAE = mean_absolute_error(y_knn_test, y_pred_knn)
MSE = mean_squared_error(y_knn_test, y_pred_knn)
RMSE = np.sqrt(mean_squared_error(y_knn_test, y_pred_knn))

print(f"R_cuadrado: {R_cuadrado:.4f}")
print(f"MAE: {MAE:.4f}")
print(f"MSE: {MSE:.4f}")
print(f"RMSE: {RMSE:.4f}")
```

```
R_cuadrado: 0.1595
MAE: 1685.3273
MSE: 4150640.3309
RMSE: 2037.3120
```

Evaluación: Una vez entrenados y obtenidas las métricas de desempeño de cada modelo, se procedió a realizar una comparación de su rendimiento para identificar el enfoque más adecuado para la predicción de la cantidad de productos a vender mensualmente. La tabla a continuación resume las métricas clave de cada modelo:

Tabla 3

Predicción de la Columna "Qty_Mensual"

Modelo	R^2	MAE	MSE	RMSE	Observaciones
Lineal	0.9168	527	410697	640	Mejor ajuste general, errores bajos y consistentes. Modelo estable y confiable.

Modelo	R^2	MAE	MSE	RMSE	Observaciones
Árbol decisión	0.9054	485	467348	683	Buen ajuste, ligeramente menor que el lineal; MAE más bajo, pero más variable.
Random Forest	0.8699	492	642238	801	Modelo robusto, pero menor precisión frente a los anteriores.
K-NN	0.1595	1685	4150640	2037	Muy bajo ajuste, errores extremadamente altos. No es adecuado para este caso.

Se evaluaron cuatro modelos de regresión para predecir la cantidad promedio mensual vendida por categoría y grupo de talla.

El modelo de regresión lineal obtuvo el mejor rendimiento global con un R^2 de 0.9168, acompañado de un bajo error absoluto medio (MAE) de 527.89, lo que sugiere un ajuste preciso y consistente del modelo a los datos.

El modelo de árbol de decisión presentó un desempeño competitivo, con un R^2 de 0.9054 y un MAE ligeramente más bajo (485.36), aunque con mayor variabilidad en los errores (RMSE = 683.63).

Por su parte, el modelo Random Forest mostró un desempeño aceptable (R^2 = 0.8699), pero con mayor error cuadrático medio en comparación con los modelos anteriores.

Finalmente, el modelo K-Nearest Neighbors (K-NN) resultó inadecuado para esta tarea, con un R^2 de apenas 0.1595 y errores significativamente altos (MAE = 1685.33, RMSE = 2037.31), lo que indica que no logra capturar correctamente la relación entre las variables predictoras y la variable objetivo.

Conclusión

Desde una perspectiva aplicada, este trabajo pone en evidencia cómo la ciencia de datos puede aportar un valor significativo a la toma de decisiones estratégicas en el manejo de inventarios. A través del análisis de datos históricos de ventas, la aplicación de modelos predictivos y el uso de técnicas de clasificación y regresión, es posible anticipar el comportamiento de la demanda con mayor precisión. Esto permite a las organizaciones optimizar la planificación de sus inventarios, ajustando la cantidad de productos disponibles según las tendencias de consumo observadas por categoría y talla.

Con el objetivo de estimar la cantidad promedio de unidades vendidas mensualmente y predecir la demanda de artículos de ropa clasificados en tallas (pequeña, mediana o grande), se aplicaron diversos modelos de regresión y clasificación. Entre los modelos utilizados se incluyen Regresión Lineal, Regresión Logística, Árboles de Decisión, Random Forest y K-Nearest Neighbors. El análisis se desarrolló sobre un conjunto de datos de ventas de Amazon del año 2022 a estados de la India. Esta modelización busca generar datos reales para la toma de decisiones estratégicas relacionadas con la gestión eficiente del inventario.

Predicción de la columna “Category”: El modelo Random Forest presentó el mejor desempeño en términos de precisión y estabilidad, demostrando una alta capacidad para manejar el desbalance de clases presente en los datos. Por su robustez y escalabilidad, se recomienda este modelo para aplicaciones prácticas en la predicción de la categoría de productos.

Predicción de la columna “Size_Agrupada”: Si bien el desempeño general de los modelos fue limitado, dado que ninguno de los modelos evaluados alcanza un rendimiento óptimo, el modelo K-Nearest Neighbors (KNN) destacó ligeramente sobre los demás, mostrando una mejor precisión general. No obstante, debido a la complejidad de predecir correctamente el grupo de

tallas, se recomienda considerar mejoras adicionales como: recolección de más datos, ingeniería de características, o técnicas de balanceo de clases para futuras versiones del modelo.

Predicción de la columna “Qty_Mensual”: El modelo de regresión lineal obtuvo el mejor desempeño en términos de ajuste y precisión general, alcanzando un coeficiente de determinación cercano a uno y errores bajos en todas las métricas evaluadas. Esto indica que el modelo logra explicar adecuadamente la variabilidad de la cantidad vendida a partir de las variables independientes escogidas. Aunque el modelo de árbol de decisión también mostró un rendimiento competitivo, el modelo lineal destaca por su simplicidad, interpretabilidad y estabilidad. En consecuencia, se recomienda como la mejor opción para predecir la cantidad de unidades vendidas mensuales, especialmente en contextos donde se requiere una estimación confiable.

El uso de herramientas de ciencia de datos, como el aprendizaje supervisado, no solo mejora la capacidad de respuesta ante cambios en el mercado, sino que también ayuda a mitigar riesgos operacionales, como el desabastecimiento de productos populares o el sobrestock de artículos de baja rotación. En consecuencia, se fortalecen aspectos clave de la gestión logística, como la eficiencia en la cadena de suministro, la reducción de costos operativos y la mejora en la experiencia del cliente al garantizar la disponibilidad de los productos más demandados.

Recomendaciones

A partir del análisis de los modelos aplicados a las tres variables objetivo, se proponen las siguientes recomendaciones generales:

Priorizar el uso de modelos de ensamble como Random Forest para problemas de clasificación con desbalance de clases, debido a su robustez y buen desempeño general.

Implementar técnicas de ingeniería de características y normalización de datos para mejorar la capacidad predictiva de los modelos.

Aplicar métodos de balanceo de clases (como SMOTE o submuestreo) para optimizar el rendimiento y evitar sobreajuste.

Realizar ajuste de hiperparámetros y validación cruzada para optimizar el desempeño de los modelos seleccionados.

En futuras etapas, considerar el uso de modelos más complejos, como redes neuronales o enfoques basados en series de tiempo, especialmente si se dispone de datos temporales más amplios.

Finalmente, se identifica como una de las limitaciones más significativas del análisis la escasez de variables explicativas. Por ello, se recomienda explorar e incorporar nuevas fuentes de información que puedan influir directamente en las ventas.

Referencias

Alteryx. (s. f.). *Glosario de ciencia de datos y análisis de datos*. Recuperado 14 de enero de 2024, de https://www.alteryx.com/es/glossary

Amazon Web Services. (s. f.). *¿Qué es Python?* Amazon Web Services. Recuperado 15 de enero de 2025, de https://aws.amazon.com/es/what-is/python/

Arimetrics. (s. f.). *Glosario Digital—Diccionario de Marketing Digital*. Arimetrics. Recuperado 15 de enero de 2025, de https://www.arimetrics.com/glosario-digital

Artem Makarov & Dmitry Namiot. (2023). Descripción general de los métodos de limpieza de datos para el aprendizaje automático. *Revista Internacional de Tecnologías de la Información Abiertas, Vol 11, No 10, 70-78*. https://cyberleninka.ru/article/n/obzor-metodov-ochistki-dannyh-dlya-mashinnogo-obucheniya

Cevallos Guamán, E. J., Jacho Gallo, A. K., & Córdova Vaca, A. M. (2024). Big Data y Analítica Predictiva en la Toma de Decisiones Empresariales. *Revista Ingenio global*, *3*(2), 55-72. https://doi.org/10.62943/rig.v3n2.2024.103

Chicaiza, S. A. A., Vaca, M. J. N., Cajo, I. M. H., & Saul, Y. P. (2024). *Métodos Avanzados para Ventas y Operaciones: Gestión Predictiva con Excel, RStudio y Python: Advanced Methods for Sales and Operations: Predictive Management with Excel, RStudio and Python*. Editorial Investigativa Latinoamericana (SciELa). https://books.google.com.co/books?id=EsLsEAAAQBAJ

Daniel Álvarez Gil. (2021, enero 14). *Metodología CRISP-DM*. Adictos al trabajo. https://adictosaltrabajo.com/2021/01/14/metodologia-crisp-dm/

Huo, Z. (2021). Sales Prediction based on Machine Learning. *2021 2nd International Conference on E-Commerce and Internet Technology (ECIT)*, 410-415. https://doi.org/10.1109/ECIT52743.2021.00093

Llatas Fernandez, L. E. (2024). Problemas de gestión de inventario que afectan la dinámica de las empresas. *Universidad César Vallejo*. https://hdl.handle.net/20.500.12692/148304

Parkash, V. (2023). A Mathematical Viewpoint on Regression Modelling of Big Data Sales Analysis using Python. *International Journal of Science and Research (IJSR), 12*(1), 1305-1310. https://doi.org/10.21275/SR23109085518

Paz Soldan Flores, Y., & Sánchez Farfán, A. D. R. (2024). *Propuestas de estrategias para gestión de inventarios en la empresa retail Wearables CO*. Universidad Peruana de Ciencias Aplicadas (UPC). http://hdl.handle.net/10757/683687

¿Qué es el bosque aleatorio? (2025, febrero 27). IBM. https://www.ibm.com/mx-es/think/topics/random-forest

Ramachandra, H. V., Balaraju, G., Rajashekar, A., & Patil, H. (2021). Machine Learning Application for Black Friday Sales Prediction Framework. *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, 57-61. https://doi.org/10.1109/ESCI50559.2021.9396994

Regression Metrics. (2024, julio 29). GeeksforGeeks. https://www.geeksforgeeks.org/regression-metrics/

Rey Escobar, I., & Valle Nieto, J. E. (2024). *Transformación digital en la logística internacional: Estrategias y desafíos de la inteligencia artificial para los inventarios y cadena de suministro en las empresas exportadoras colombianas*. http://hdl.handle.net/20.500.11912/12163

Roberto Díaz Badra. (2020, agosto 5). *Métricas de Clasificación*. The Machine Learners. https://www.themachinelearners.com/metricas-de-clasificacion/

Sai Nikhil Boyapati & Ramesh Mummidi. (2020). *Predicción de ventas mediante máquinas Técnicas de aprendizaje* [Instituto Tecnológico de Blekinge]. https://www.diva-portal.org/smash/get/diva2:1455353/FULLTEXT02

Subhendu Kumar Pani, Abhaya Kumar Samal, Kabita Sahoo, & Jitendra Pramanik. (2019). Exploratory Data Analysis using Python. *International Journal of Innovative Technology and Exploring Engineering*, *8*(12), 4727-4735. https://doi.org/10.35940/ijitee.L3591.1081219

Velasco Rebolledo, J. (2024). *Machine Learning: Fundamentos, Algoritmos y Aplicaciones para Los Negocios, Industria y Finanzas* (1st ed). Ediciones Diaz de Santos S.A. https://www.google.com.co/books/edition/Machine_learning/eMYpEQAAQBAJ?hl=es-419&gbpv=0

Yajure Ramírez, C. A. (2022). Aplicación de la metodología de Ciencia de Datos para analizar datos de facturación de energía eléctrica. Caso de estudio: Uruguay 2000-2022. *Revista de investigación de Sistemas e Informática*, *15*(1), 127-138. https://doi.org/10.15381/risi.v15i1.23544

Yi, S. (2023). Walmart Sales Prediction Based on Machine Learning. *Highlights in Science, Engineering and Technology*, *47*, 87-94. https://doi.org/10.54097/hset.v47i.8170

Yobel SCM Corp. (2024, septiembre 25). *5 Fallos en la gestión de inventarios y soluciones en manufactura*. https://www.yobelscm.biz/landing-cr/fallos-en-gestion-de-inventarios/

Apéndice

Apéndice A

Regresión Logística para Predicción de "Size_Agrupada"

```
df_ml = df.copy()

y_ml = df_ml['Grupo_Talla']

x_ml = df_ml[['Category', 'ship-state', 'ship-service-level', 'Fulfilment', 'Qty', 'Amount', 'B2B']]
x_ml = pd.get_dummies(x_ml, columns=['Category'], drop_first=True)

x_ml_train, x_ml_test, y_ml_train, y_ml_test = train_test_split(x_ml, y_ml, test_size=0.2, random_state=45)

mod_ml = make_pipeline(StandardScaler(), LogisticRegression(max_iter=2000, class_weight='balanced', random_state=45))

scores_ml = cross_validate(mod_ml, x_ml, y_ml, cv=5, scoring=['accuracy'])
print("Accuracy promedio:", scores_ml['test_accuracy'].mean())

mod_ml.fit(x_ml_train, y_ml_train)
y_ml_pred = mod_ml.predict(x_ml_test)
print("\nReporte de clasificación:\n", classification_report(y_ml_test, y_ml_pred, zero_division=0))
```

Apéndice B

Árbol de Decisión para Predicción de "Size_Agrupada"

```
df_ad = df.copy()

y_ad = df_ad['Grupo_Talla']

x_ad = df_ad[['Category', 'ship-state', 'ship-service-level', 'Fulfilment', 'Qty', 'Amount', 'B2B']]
x_ad = pd.get_dummies(x_ad, columns=['Category'], drop_first=True)

x_ad_train, x_ad_test, y_ad_train, y_ad_test = train_test_split(x_ad, y_ad, test_size=0.2, random_state=45)

mod_ad = make_pipeline(StandardScaler(), DecisionTreeClassifier(max_depth=6, class_weight='balanced', random_state=45))

scores_ad = cross_validate(mod_ad, x_ad, y_ad, cv=5, scoring=['accuracy'])
print("Accuracy promedio:", scores_ad['test_accuracy'].mean())

mod_ad.fit(x_ad_train, y_ad_train)
y_ad_pred = mod_ad.predict(x_ad_test)
print("\nReporte de clasificación:\n", classification_report(y_ad_test, y_ad_pred, zero_division=0))
```

Apéndice C

Random Forest para Predicción de "Size_Agrupada"

```
df_rf = df.copy()

y_rf = df_rf['Grupo_Talla']

x_rf = df_rf[['Category', 'ship-state', 'ship-service-level', 'Fulfilment', 'Qty', 'Amount', 'B2B']]
x_rf = pd.get_dummies(x_rf, columns=['Category'], drop_first=True)

x_rf_train, x_rf_test, y_rf_train, y_rf_test = train_test_split(x_rf, y_rf, test_size=0.2, random_state=45)

mod_rf = make_pipeline(StandardScaler(),
                       RandomForestClassifier(n_estimators=200, class_weight='balanced', random_state=45))

scores_rf = cross_validate(mod_rf, x_rf, y_rf, cv=5, scoring=['accuracy'])
print("Accuracy promedio:", scores_rf['test_accuracy'].mean())

mod_rf.fit(x_rf_train, y_rf_train)
y_rf_pred = mod_rf.predict(x_rf_test)
print("\nReporte de clasificación:\n", classification_report(y_rf_test, y_rf_pred))
```

Apéndice D

KNN para Predicción de "Size_Agrupada"

```
df_knn = df.copy()

y_knn = df_knn['Grupo_Talla']

x_knn = df_knn[['Category', 'ship-state', 'ship-service-level', 'Fulfilment', 'Qty', 'Amount', 'B2B']]
x_knn = pd.get_dummies(x_knn, columns=['Category'], drop_first=True)

x_knn_train, x_knn_test, y_knn_train, y_knn_test = train_test_split(x_knn, y_knn, test_size = 0.2, random_state = 45)

mod_knn = make_pipeline(StandardScaler(), KNeighborsClassifier(n_neighbors=5))

scores_knn = cross_validate(mod_knn, x_knn, y_knn, cv=5, scoring=['accuracy'])
print("Accuracy promedio:", scores_knn['test_accuracy'].mean())

mod_knn.fit(x_knn_train, y_knn_train)
y_knn_pred = mod_knn.predict(x_knn_test)
print("\nReporte de clasificación:\n", classification_report(y_knn_test, y_knn_pred))
```