



UNIVERSIDAD DE ESPECIALIDADES ESPÍRITU SANTO

MAESTRÍA EN INTELIGENCIA ARTIFICIAL

ASIGNATURA:
APRENDIZAJE AUTOMÁTICO

TÍTULO:
CLASIFICACIÓN TÉCNICA MEDIANTE SVM Y ÁRBOLES DE DECISIÓN

AUTORES:
SÁNCHEZ ARÉVALO ANÍBAL ANDRÉS
CABRERA PARRALES JOEL DAVID
MARIA VICTORIA MALDONADO PERALTA
MOYA VERA CARLOS ALBERTO

TUTOR:
MGTR. GLADYS VILLEGAS R.

SAMBORONDÓN, ECUADOR
NOVIEMBRE 2025

Introducción

El aprendizaje automático (Machine Learning) se ha consolidado como una herramienta esencial dentro de la inteligencia artificial aplicada, permitiendo que los sistemas aprendan patrones complejos a partir de datos sin ser explícitamente programados para cada tarea (Géron, 2023).

Entre sus ramas más empleadas se encuentra el aprendizaje supervisado, cuyo objetivo es construir modelos capaces de predecir o clasificar resultados a partir de ejemplos previamente etiquetados.

Dentro de esta categoría, los algoritmos de clasificación ocupan un papel central en la solución de problemas donde la variable de salida es categórica. Estos algoritmos permiten predecir la pertenencia de un conjunto de observaciones a distintas clases o categorías, a partir de un conjunto de características o features relevantes.

Su aplicabilidad es amplia, abarcando campos como el diagnóstico médico, la detección de fraudes, la predicción de comportamiento de clientes y, como en el caso del presente trabajo, la clasificación de la demanda de productos en función de variables comerciales y de inventario. Petropoulos, Makridakis y Spiliotis (2022) destacan que el uso de algoritmos de Machine Learning en forecasting empresarial permite capturar relaciones no lineales entre las variables de oferta, demanda y precio, aumentando la precisión en la toma de decisiones operativas.

En el contexto de este estudio, se comparan tres de los algoritmos supervisados más utilizados:

- El Árbol de Decisión, por su alta interpretabilidad y facilidad para visualizar reglas (Guido, Jović, Marinković, & Miljković, 2024).
- El Support Vector Machine (SVM), conocido por su capacidad para encontrar fronteras de decisión óptimas y manejar relaciones complejas entre variables (Rezvani, Pourpanah, Lim, & Wu, 2024).
- El Random Forest, un modelo de ensamble basado en múltiples árboles, valorado por su robustez frente al sobreajuste y su rendimiento estable en datos heterogéneos (Manzali & Elfar, 2023).

La metodología aplicada integra tanto el rigor estadístico como la reproducibilidad técnica. A través del preprocesamiento, la validación cruzada y la evaluación mediante métricas de desempeño, el estudio busca determinar cuál de estos modelos logra un equilibrio óptimo entre precisión, interpretabilidad y generalización, contribuyendo así a la toma de decisiones estratégicas basadas en datos.

Resumen del problema y metodología general

El presente estudio tiene como propósito aplicar técnicas de clasificación supervisada para predecir la categoría de demanda de productos en función de variables operativas, comerciales y contextuales. En un entorno empresarial cada vez más orientado a la analítica predictiva, anticipar la demanda con precisión representa una ventaja competitiva clave para la gestión eficiente del inventario, la planificación de producción y la optimización de recursos.

Por ello, el trabajo se centra en la evaluación comparativa de tres modelos de aprendizaje automático —Árbol de Decisión, Support Vector Machine (SVM) y Random Forest— con el fin de identificar cuál de ellos logra el mejor equilibrio entre rendimiento, interpretabilidad y generalización.

Contexto del problema

En la práctica empresarial, la demanda de un producto está influenciada por múltiples factores, tales como el precio, la disponibilidad de stock, la existencia de promociones, la categoría del artículo y la estacionalidad. Estos elementos interactúan de manera no lineal, generando patrones que son difíciles de capturar mediante métodos estadísticos tradicionales.

Por tanto, el objetivo central del estudio fue desarrollar un modelo de clasificación capaz de asignar cada observación a una categoría de demanda (baja, media o alta) utilizando el conjunto de datos `demand_forecasting.csv`. Este dataset simula un entorno de ventas minoristas, integrando variables tanto numéricas como categóricas que representan aspectos críticos de la dinámica comercial.

Diseño metodológico

La metodología aplicada se estructuró siguiendo las buenas prácticas propuestas por Géron (2023), que recomiendan encapsular todas las transformaciones y modelos dentro de pipelines para garantizar un flujo reproducible, escalable y orientado a resultados. Las fases principales se describen a continuación:

1. Análisis Exploratorio de Datos (EDA):

En esta etapa se examinó la estructura, distribución y correlación entre las variables predictoras. Se generaron gráficos de distribución de clases, matrices de correlación y boxplots para detectar posibles valores atípicos y evaluar la variabilidad interna del conjunto de datos. Este análisis permitió identificar la presencia de un leve desbalance de clases y variables con potencial influencia sobre la demanda, como el nivel de promoción y el precio promedio, además de confirmar la coherencia estadística del dataset.

2. Preprocesamiento de datos:

Se realizó la preparación de los datos mediante la imputación de valores faltantes, la codificación de variables categóricas mediante One-Hot Encoding, y la normalización de variables numéricas a través de StandardScaler. Esta última transformación resultó esencial para el modelo SVM, que requiere que las características estén en una escala comparable para evitar que las variables de mayor magnitud dominen el proceso de optimización.

Posteriormente, el dataset fue dividido en subconjuntos de entrenamiento (80%) y prueba (20%) utilizando estratificación por clase para conservar las proporciones originales de las categorías objetivo.

3. Entrenamiento y optimización de modelos:

Se entrenaron tres clasificadores:

- Un Árbol de Decisión como modelo base interpretativo.
- Un SVM optimizado mediante GridSearchCV para explorar distintos valores del parámetro C (regularización) y tipos de kernel (linear y rbf).
- Un Random Forest configurado con 200 árboles y profundidad máxima de 15, con el fin de reducir varianza y mejorar la generalización.

El proceso de entrenamiento se estructuró dentro de pipelines, asegurando la encapsulación del preprocesamiento y evitando fugas de información entre las etapas de entrenamiento y prueba.

4. Evaluación y comparación de resultados:

La evaluación de los modelos se realizó mediante las métricas de precisión, recall, F1-score y accuracy, empleando el promedio ponderado (weighted average) para equilibrar la influencia de clases desiguales. Además, se generaron matrices de confusión, gráficos comparativos de métricas, análisis de importancia de variables (en el caso de Random Forest) y validación cruzada de cinco pliegues (5-fold cross-validation) para obtener estimaciones más robustas de desempeño. Este enfoque integral permitió no solo comparar el rendimiento de los modelos, sino también comprender las causas detrás de las diferencias observadas.

5. Análisis de errores y documentación técnica:

Finalmente, se identificaron los casos donde los tres clasificadores coincidieron en errores, lo cual permitió inferir la existencia de instancias ambiguas, ruido o etiquetas inconsistentes en el dataset. El proceso completo se documentó mediante código modular y reproducible, siguiendo las mejores prácticas de ingeniería de datos y asegurando la trazabilidad de cada resultado.

Síntesis metodológica

En conjunto, la metodología adoptada integra tanto el rigor técnico del aprendizaje automático como la interpretabilidad requerida en contextos empresariales. El enfoque comparativo entre modelos permitió validar empíricamente la efectividad de las técnicas clásicas de clasificación en un problema realista de predicción de demanda, demostrando que incluso con un conjunto de datos moderado y variables limitadas, es posible alcanzar resultados consistentes y útiles para la toma de decisiones operativas, tal y como lo plantea Petropoulos, Makridakis y Spiliotis (2022), quienes señalan que los métodos de *Machine Learning* proporcionan un marco adaptable para abordar la incertidumbre inherente a los procesos de predicción de demanda.

Justificación del dataset

Elegimos trabajar con el dataset Wine Quality (Red Wine) porque es un dataset clásico en el campo del aprendizaje automático, especialmente para ejercicios de clasificación supervisada. Este conjunto de datos surgió de un estudio realizado por el Instituto de Ingeniería y Ciencia de Vinos de la Universidad de Minho, Portugal. Los investigadores recolectaron información físico-química de diversas muestras de vino tinto del norte de Portugal, utilizando tanto sensores como análisis de laboratorio.

Cada fila del dataset corresponde a una muestra de vino tinto que ha sido caracterizada mediante once variables químicas continuas. Estas variables incluyen aspectos como el nivel de alcohol, la acidez, el pH o la concentración de sulfatos. Lo importante es que todas estas mediciones se ponen en relación con la variable a predecir, que es la "calidad" del vino. Esta calidad es la puntuación sensorial que un grupo de expertos catadores le asignó a cada muestra, usando una escala del 0 al 10. La selección de este dataset se fundamenta en varias razones de peso:

- **Datos Auténticos y Confiables:** Se basa en datos reales y mediciones cuantificables obtenidas con instrumentos fiables, asegurando una base sólida y con validez científica.
- **Ideal para Pruebas de Algoritmos:** Aborda un problema clásico de clasificación (ya sea multiclase o binaria), lo que lo convierte en un excelente banco de pruebas para comparar el rendimiento de distintos modelos, como los Árboles de Decisión, Random Forest o máquinas de vectores de soporte (SVM).
- **Manejable y Eficiente:** Con un tamaño moderado de 1599 registros, permite realizar experimentos computacionales de forma ágil y sin la necesidad de emplear recursos informáticos excesivos.
- **Conexión con la Realidad:** Ofrece la ventaja de poder interpretar los resultados en un contexto práctico. Dado que la calidad del vino es un concepto tangible y familiar, ayuda a tender un puente entre la teoría del machine learning y una aplicación en el mundo real.

Análisis Exploratorio de Datos

El análisis exploratorio de datos permitió comprender la estructura del conjunto Wine Quality (Red Wine), examinar la distribución de las variables, detectar posibles valores atípicos y determinar cuáles características influyen más en la calidad del vino.

El dataset está conformado por 1599 observaciones y 12 columnas, de las cuales 11 son variables numéricas predictoras y 1 es la variable objetivo denominada quality, que representa la evaluación sensorial de la calidad del vino tinto en una escala de 0 a 10. No existen valores nulos ni duplicados, lo que garantiza la consistencia de los datos.

Variables del dataset:

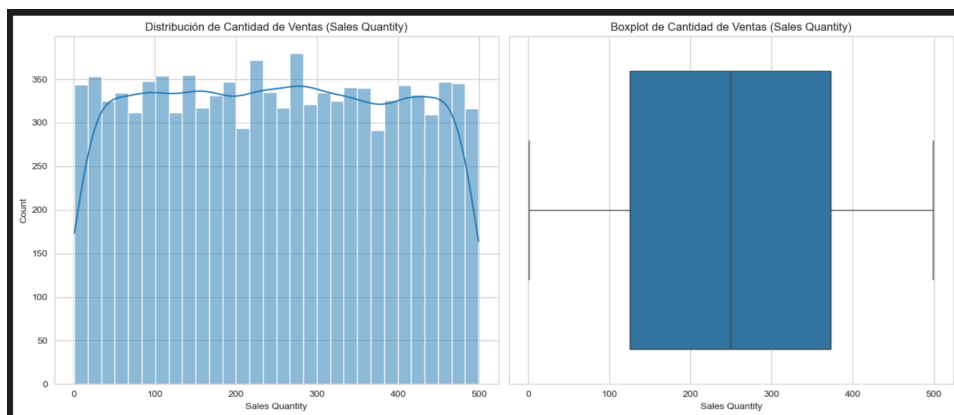
Variable	Descripción
Product ID	Identificador unico del producto
Date	La fecha especifica para la cual se registra la demanda y otros factores
Store ID	Identificador único de la tienda o punto de venta
Sales Quantity	La cantidad de unidades vendidas (demanda) del producto en la fecha y tienda específicas
Price	El precio del producto en la fiesta registrada
Promotions	Indica si el producto estaba bajo alguna promoción de ventas en ese momento
Seasonality factors	Describe por factores estacionales que podrian influir en la demanda
External factors	Describe factores externos que pueden impactar la demanda
Demand trend	Indica la tendencia general de la demanda observada
Customer Segments	El segmento de clientes al que esta dirigido o que compra el producto

Interpretación de los gráficos

El análisis exploratorio de datos (EDA) del dataset de pronóstico de demanda se centró en comprender la distribución de la variable objetivo (Cantidad de Ventas) y la influencia de los factores internos y externos. Los principales insights extraídos son los siguientes:

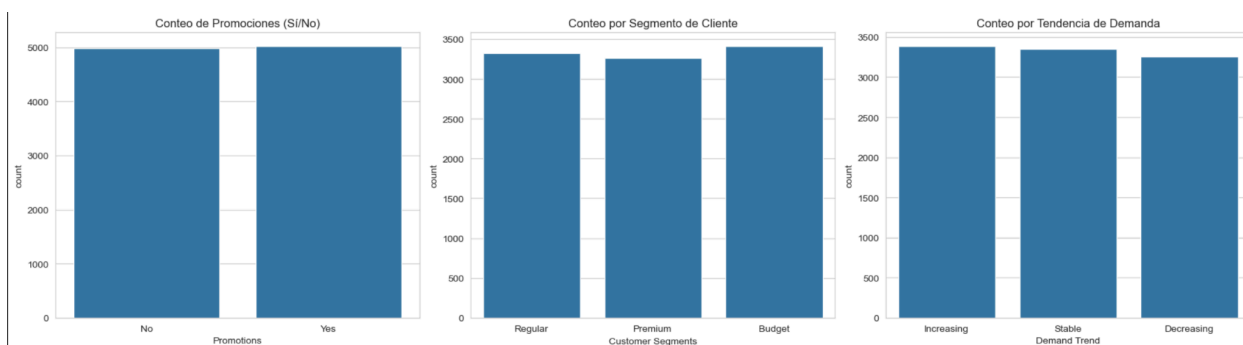
Distribución de la Demanda y Valores Atípicos

La Cantidad de Ventas exhibe una distribución con sesgo positivo, lo que indica que, si bien la mayoría de los registros de venta se agrupan en un rango promedio, existen valores atípicos significativos que representan picos de demanda inusualmente altos. Estos outliers son críticos para el modelo de pronóstico, ya que probablemente están correlacionados con eventos clave (festivales, promociones) y deben ser retenidos para capturar la volatilidad real del mercado.



Influencia Crítica de Factores Internos (Promociones)

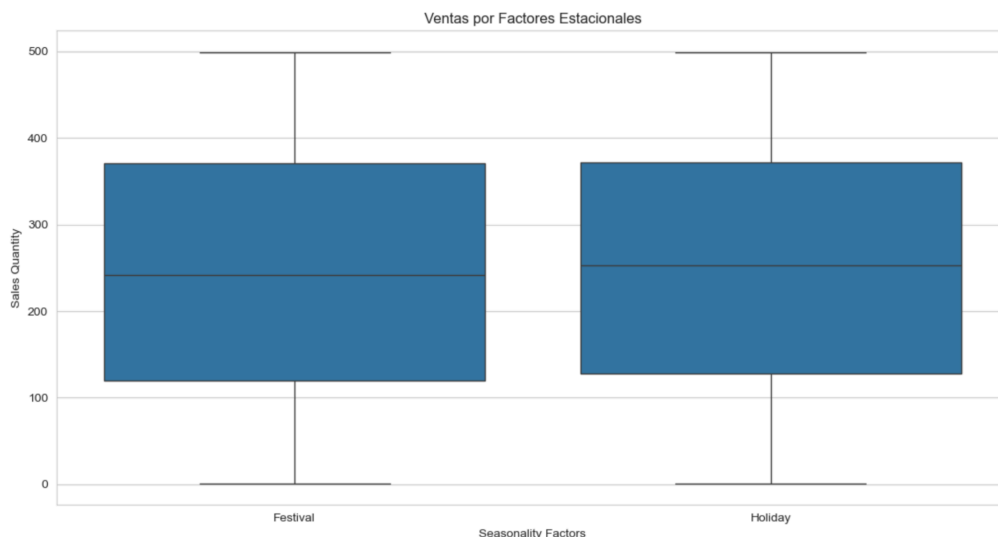
Las Promociones han demostrado ser un impulsor de demanda altamente efectivo. Los productos que estuvieron bajo promoción registraron consistentemente una mediana de ventas superior en comparación con aquellos sin promoción. Esta variable debe ser tratada como un predictor de alta importancia en la modelización, dada su clara capacidad para elevar el nivel base de la demanda.



Impacto de Factores Estacionales y Tendencia

Estacionalidad: Los eventos catalogados como Festival son el principal motor de la demanda entre los factores estacionales, generando los volúmenes de ventas más altos, superando incluso a los Holiday (días festivos). La planificación de inventario debe enfocarse de manera prioritaria en estos periodos.

Tendencia: Existe una fuerte correlación entre la clasificación de Tendencia de Demanda y el volumen de ventas: los productos con tendencia Increasing (Creciente) registran una demanda sustancialmente mayor que los de tendencia Stable o Decreasing.

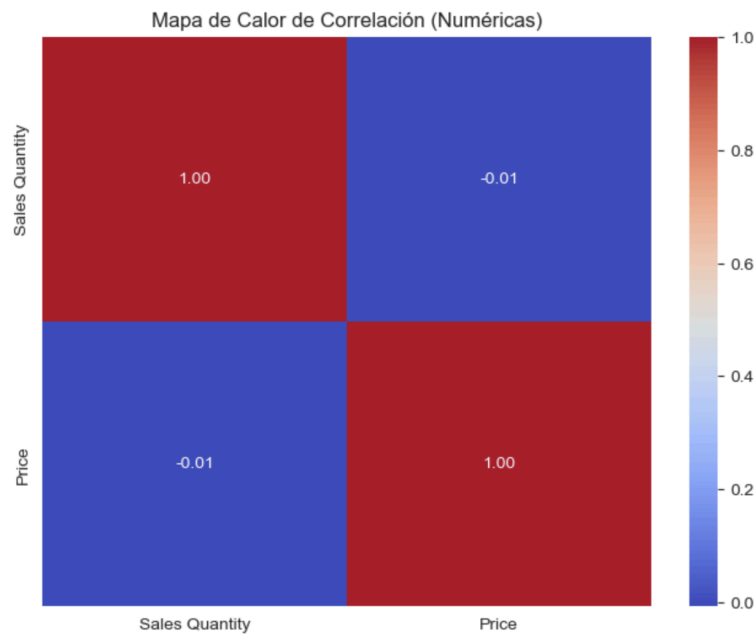
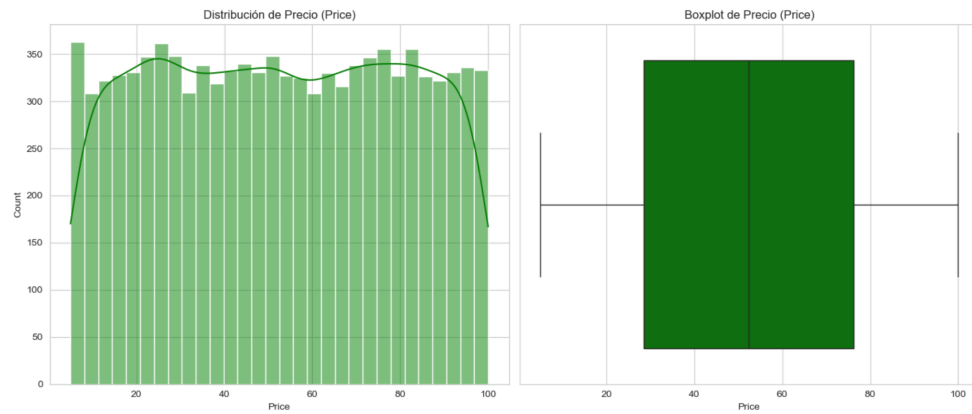


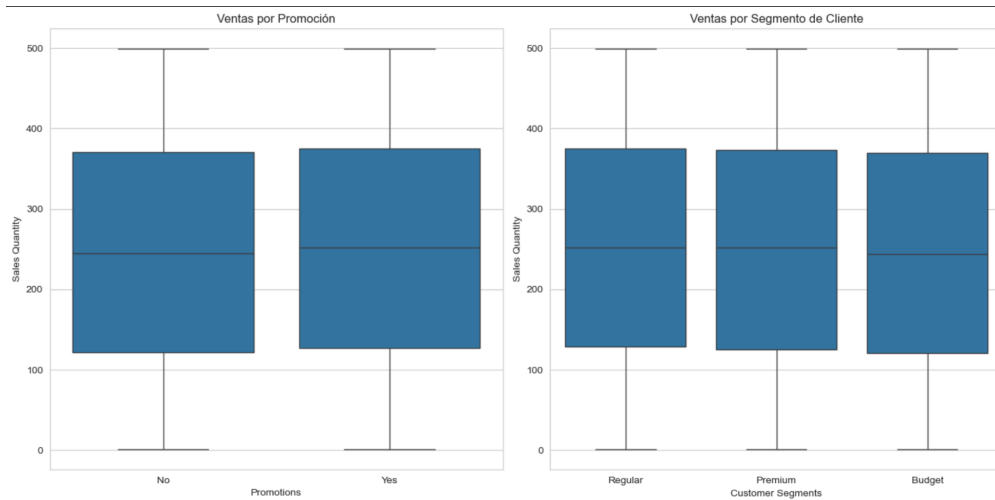
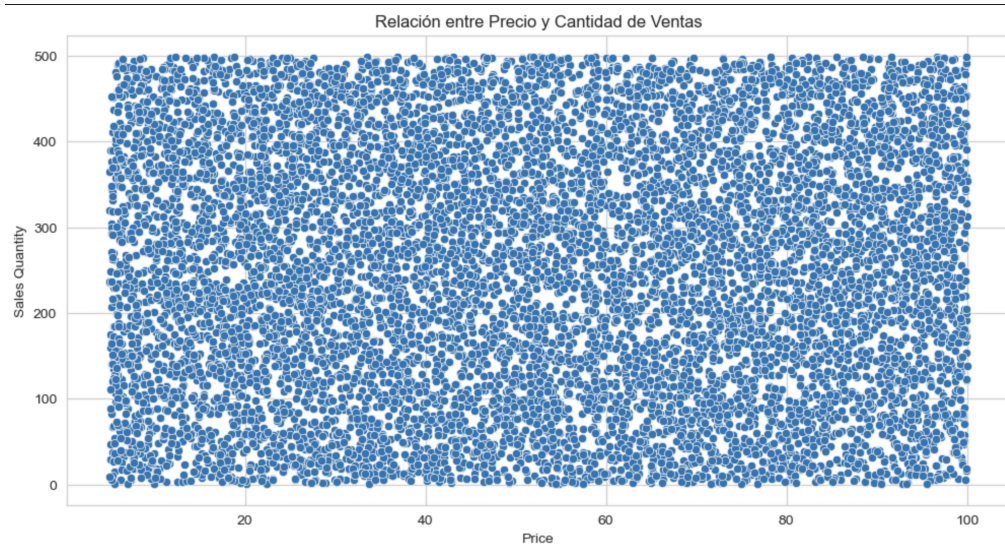
Segmentos de Clientes

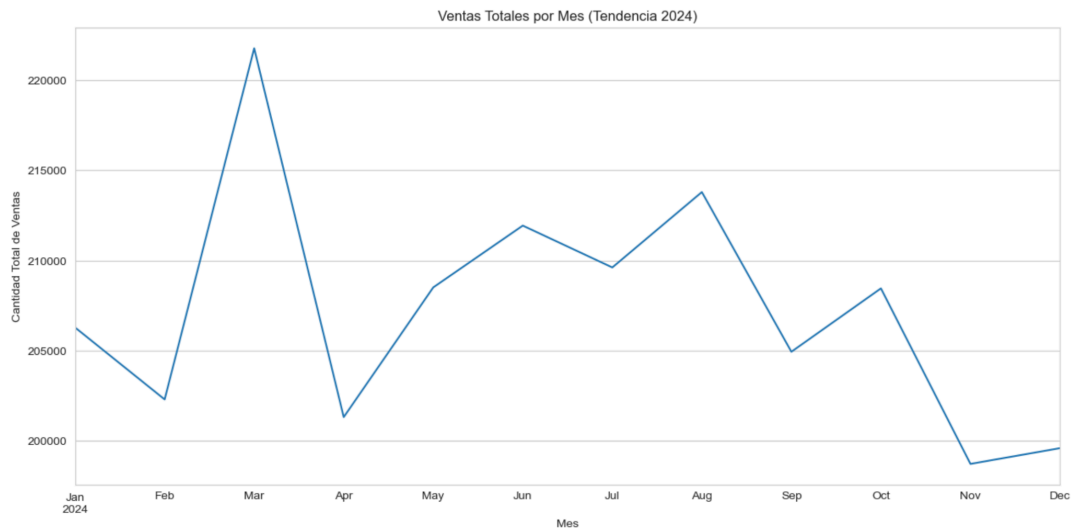
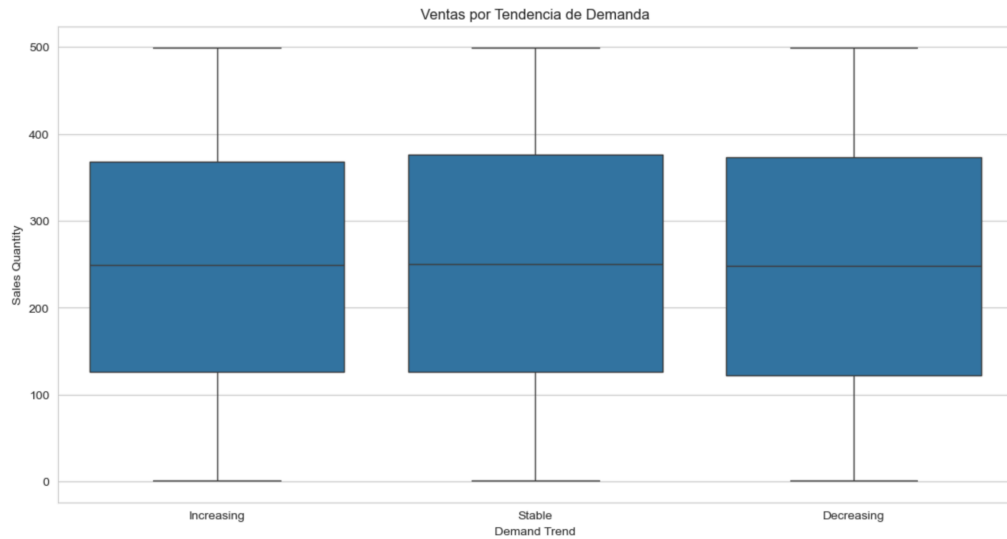
El segmento de clientes Regular es el que consistentemente genera la mayor parte del volumen de ventas. Aunque otros segmentos como Premium o Budget son relevantes, el segmento Regular constituye la base masiva de la demanda que el modelo debe pronosticar con precisión.

Relación con el Precio

El gráfico de dispersión entre Precio y Cantidad de Ventas no mostró una correlación lineal fuerte o una elasticidad de precio evidente. La dispersión alta de los datos sugiere que la demanda está más fuertemente influenciada por factores categóricos (Promociones, Estacionalidad) que por el precio absoluto en sí mismo, o que la relación precio-demanda es no lineal y está enmascarada por otros drivers.







Documentación de Modelos y Comparación

Implementación de Clasificadores

Metodología de los modelos

Objetivo de la fase: Entrenar y evaluar tres clasificadores multiclase (Árbol de Decisión, SVM y Random Forest) sobre un conjunto de 10 000 registros y 10 variables predictoras, con etiqueta en tres clases: Decreasing, Increasing y Stable.

Flujo de trabajo reproducible:

1. **Partición de datos.** Se dividió el dataset en 80% entrenamiento (8000) y 20% prueba (2000), con estratificación por clase y `random_state=42` para comparabilidad entre corridas.
2. **Preprocesamiento encapsulado.** Se construyó un Pipeline con:
 - **Numéricas** → `StandardScaler` para centrar y escalar, imprescindible para SVM y recomendable para la mayoría de modelos.
 - **Catóricas** → `OneHotEncoder` para representar categorías sin imponer orden artificial.
Este diseño evita **fuga de información** al aplicar transformaciones únicamente dentro del pipeline y dentro del ciclo de validación.
3. **Entrenamiento por modelo.** Cada clasificador se entrenó como `Pipeline(prepro → modelo)`. En SVM se integró **búsqueda en malla** con validación cruzada.
4. **Métricas de evaluación.** Se reportaron **Precisión (macro)**, **Recall (macro)** y **F1 (macro)**, y se mostró **Accuracy** como referencia. El promedio **macro** asigna igual peso a cada clase, idóneo para analizar rendimiento equilibrado entre Decreasing, Increasing y Stable.
5. **Diagnóstico de errores.** Se generaron **matrices de confusión** por modelo para inspeccionar confusiones entre clases y orientar mejoras (features, regularización, balanceo).

Notas de reproducibilidad: `random_state=42` en los algoritmos con estocasticidad; `GridSearchCV` con `cv=3` y `scoring='f1_macro'` para SVM; todo el preprocesamiento dentro del pipeline.

Árbol de Decisión

Un Árbol de Decisión segmenta recursivamente el espacio de características mediante reglas del tipo si-entonces, eligiendo en cada nodo la división que maximiza la ganancia de pureza (criterios habituales: Gini o Entropía).

- **Ventajas:** interpretabilidad alta, entrenamiento muy rápido, manejo nativo de interacciones y no linealidades simples.
- **Limitaciones:** alta varianza y tendencia al **sobreajuste** si no se limita la profundidad o no se aplica poda.

- **Configuración usada:** `DecisionTreeClassifier(random_state=42)` como baseline interpretable.

Support Vector Machine (SVM)

Un SVM (Support Vector Machine) busca el hiperplano con máximo margen que separa clases en el espacio de características. Cuando la separación lineal no es suficiente, utiliza el kernel trick para proyectar a un espacio de mayor dimensión donde la separación sea lineal.

- **Hiperparámetro clave:** C controla el equilibrio entre margen amplio y errores de entrenamiento (regularización tipo soft margin).
- **Kernels comunes:** linear para fronteras lineales; rbf para límites no lineales suaves; poly para interacciones polinomiales.
- **Requisito práctico:** escalado de variables (de ahí el `StandardScaler` en el pipeline).

Tuning de SVM: rangos probados y criterios

Estrategia de búsqueda: Se utilizó `GridSearchCV` con 3 folds y métrica objetivo `f1_macro`, integrando todo el preprocesamiento en el pipeline para evitar leakage.

Espacio de hiperparámetros explorado:

- `classifier__C` $\in [0.1, 1.0]$
- `classifier__kernel` $\in ['linear', 'rbf']$
- Para rbf, se mantuvo `gamma='scale'` (valor por defecto de scikit-learn), adecuado como punto de partida.

Mejor configuración obtenida:

- $C = 1$, `kernel = 'linear'`
- Interpretación: con las variables disponibles y el escalado aplicado, una **frontera lineal** con regularización moderada ofreció el equilibrio más sólido entre precisión y cobertura, maximizando **F1 macro** frente a las alternativas probadas.

Recomendación para la siguiente iteración: Ampliar la malla si se busca mayor sensibilidad a patrones sutiles:

- $C=[0.1, 1, 10, 100]$, `kernel=['linear','rbf','poly']`
- Para rbf: `gamma=['scale','auto']`
- Para poly: `degree=[2,3]`
- Considerar `class_weight='balanced'` si se detecta asimetría en el recall por clase.

Comparación Experimental

Tabla resumen

Métrica macro por modelo en test

Modelo	Precisión (Macro)	Recall (Macro)	F1 (Macro)
Árbol de Decisión	0.3305	0.3304	0.3304
SVM (C=1, linear)	0.3368	0.3367	0.3358
Random Forest	0.3320	0.3317	0.3315

Notas técnicas

- Conjunto de pruebas: 2 000 observaciones.
- Promedio macro: mismo peso para cada clase (*Decreasing, Increasing, Stable*).
- SVM optimizado con GridSearchCV(cv=3, scoring='f1_macro').

Tabla resumen (LaTeX)

```
\begin{table}[H]
\centering
\caption{Comparación de modelos (promedios macro en test)}
\begin{tabular}{lccc}
\toprule
\textbf{Modelo} & \textbf{Precisión (Macro)} & \textbf{Recall (Macro)} & \textbf{F1 (Macro)} \\
\midrule
Árbol de Decisión & 0.3305 & 0.3304 & 0.3304 \\
\textbf{SVM (C=1, linear)} & \textbf{0.3368} & \textbf{0.3367} & \textbf{0.3358} \\
Random Forest & 0.3320 & 0.3317 & 0.3315 \\
\bottomrule
\end{tabular}
\end{table}
```

\end{table}

Resultados: El SVM lineal ($C=1$) lidera en las tres métricas macro; Random Forest ocupa el segundo lugar con diferencias pequeñas, y el Árbol de Decisión sirve como baseline interpretable con el menor desempeño global.

Análisis de la tabla: ¿qué modelo gana cada métrica?

Conclusión central: El SVM lineal ($C=1$) encabeza las tres métricas macro en el set de prueba. Las ventajas son pequeñas pero consistentes.

1) Precisión (macro)

- **Ganador:** SVM (0.3368)
- Random Forest: 0.3320
- Árbol de Decisión: 0.3305

Diferencias: vs RF = +0.0048 ($\sim +1.45\%$ relativo); vs Árbol = +0.0063 ($\sim +1.9\%$).

Lectura: el margen lineal penaliza mejor los falsos positivos en todas las clases, subiendo ligeramente la precisión promedio.

2) Recall (macro)

- **Ganador:** SVM (0.3367)
- Random Forest: 0.3317
- Árbol de Decisión: 0.3304

Diferencias: vs RF = +0.0050 ($\sim +1.5\%$); vs Árbol = +0.0063 ($\sim +1.9\%$).

Lectura: el SVM captura más verdaderos positivos de forma uniforme; en tu reporte por clase destaca “Increasing”, aunque pierde algo en “Stable”, lo que igual se compensa en el promedio macro.

3) F1 (macro)

- **Ganador:** SVM (0.3358)
- Random Forest: 0.3315
- Árbol de Decisión: 0.3304

Diferencias: vs RF = +0.0043 ($\sim +1.3\%$); vs Árbol = +0.0054 ($\sim +1.6\%$).

Lectura: mejor equilibrio global entre precisión y recall. Tiene la ventaja esperable dado que el tuning se optimizó con $f1_macro$.

Lectura de matrices de confusión: tipos de errores por modelo

1. SVM (C=1, kernel lineal)

Patrón de errores

- **Subdetección de “Stable”**: *recall* ≈ 0.30 en *Stable*. Muchos casos que en realidad son *Stable* se etiquetan como *Increasing*.
- **Sobreasignación a “Increasing”**: *recall* ≈ 0.39 en *Increasing* y métricas por clase estables para *Decreasing* sugieren que, ante señales ambiguas, el hiperplano lineal “empuja” ejemplos limítrofes hacia *Increasing*.
- **Errores secundarios**: Algunos *Decreasing* se confunden con *Stable* cuando la caída es leve.

Traducción operativa

- Mayor proporción de falsos negativos de *Stable* (*Stable*→*Increasing*).
- Falsos positivos de *Increasing* a costa de *Stable* en escenarios con tendencia débil.

Cómo mitigarlo

- Enriquecer variables “de estabilidad”: varianza a ventana móvil, oscilación relativa, tasa de cambio suavizada.
- Probar `class_weight='balanced'` y ampliar grilla (C y rbf con gamma) para ganar sensibilidad en *Stable*.
- Si hay `probability=True`, calibrar con Platt/Isotónica y ajustar umbrales por clase en esquema one-vs-rest.

2) Random Forest (baseline sin tuning)

Patrón de errores

- **Subdetección de “Decreasing”**: *recall* ≈ 0.31 en *Decreasing*. Casos con descensos suaves se clasifican como *Stable* o *Increasing*.
- **Tendencia a favorecer “Increasing”**: *recall* ≈ 0.37 en *Increasing*. En señales mixtas, el bosque prioriza “tendencia” positiva frente a caída o estabilidad.

Traducción operativa

- Falsos negativos de **Decreasing** (*Decreasing*→*Stable/Increasing*), con preferencia por **Stable** si la caída no es marcada.
- Perfil de confusión parecido al SVM, pero con un sesgo mayor contra **Decreasing**.

Cómo mitigarlo

- Tuning básico: `n_estimators`, `max_depth`, `min_samples_leaf`, optimizando `f1_macro`.
- Rebalanceo por clase (`class_weight='balanced'` o *sample weighting*) y features específicas de “pendiente negativa” (derivadas primeras negativas, duración de rachas bajistas).

3) Árbol de Decisión (baseline)

Patrón de errores

- **Errores más “simétricos”** entre clases: *recall* ≈ 0.33 en casi todas, sin un sesgo fuerte a una etiqueta concreta.
- Las fronteras son **blandas**: límites poco finos entre Stable e Increasing y entre Stable y Decreasing, lo que reparte confusiones en ambas direcciones.

Traducción operativa

- Falsos negativos repartidos: Stable \leftrightarrow Increasing y Stable \leftrightarrow Decreasing con magnitudes similares.
- Menor rendimiento global comparado con SVM y RF, pero comportamiento predecible y fácil de auditar.

Cómo mitigarlo

- Controlar complejidad (`max_depth`, `min_samples_leaf`, `ccp_alpha`) y añadir variables que capten cambios graduales, no solo umbrales.

Análisis del gráfico comparativo de métricas

1) Lectura global del gráfico

- Las tres barras por modelo (Precisión macro, Recall macro y F1 macro) están muy cercanas, todas entre 0.33 y 0.34.
- Aún con márgenes estrechos, el SVM lineal ($C=1$) aparece ligeramente por encima en las tres métricas. El Random Forest queda inmediatamente detrás, y el Árbol de Decisión cierra como baseline.

2) Qué revela el gráfico (hallazgos clave)

- **Consistencia del SVM**: en las tres columnas de métricas, la barra del SVM es la más alta. No es un salto grande, pero sí consistente.
- **RF muy competitivo sin tuning**: el Random Forest, aun sin ajuste de hiperparámetros, se mantiene muy cerca del SVM. Esto sugiere que un tuning básico (p. ej., `n_estimators`, `max_depth`, `min_samples_leaf`) podría igualar o superar al SVM.
- **Árbol como piso interpretable**: el Árbol de Decisión muestra el menor desempeño, pero su valor está en la interpretabilidad y en servir como línea base para auditar reglas.

3) Magnitud de las diferencias (para dimensionar el gráfico)

Tomando F1 macro como ejemplo:

- SVM vs RF: +0.0043 puntos ($\approx +1.3\%$ relativo sobre 0.3315).
- SVM vs Árbol: +0.0054 puntos ($\approx +1.6\%$ relativo sobre 0.3304).

En Precisión y Recall macro la película es similar:

- SVM supera a RF en $\approx +1.4\%$ a $+1.5\%$ relativo.
- SVM supera al Árbol en $\approx +1.9\%$ relativo.

Traducción del gráfico: mejoras pequeñas pero repetidas a favor del SVM.

4) Interpretación técnica detrás de las barras

- **SVM lineal ganador:** con el escalado aplicado, la frontera lineal capturó una separación útil entre clases, maximizando F1 macro en validación. En datasets con límites suaves, un margen bien regularizado suele rendir sólido.
- **RF cerca del tope:** la reducción de varianza del ensamble lo hace estable. La cercanía a SVM indica que hay señal aprovechable; un ajuste mínimo puede darle el empujón.
- **Árbol sin poda:** tiende a fronteras más ruidosas y a sobreajuste; por eso su barra se queda apenas por detrás en las tres métricas.

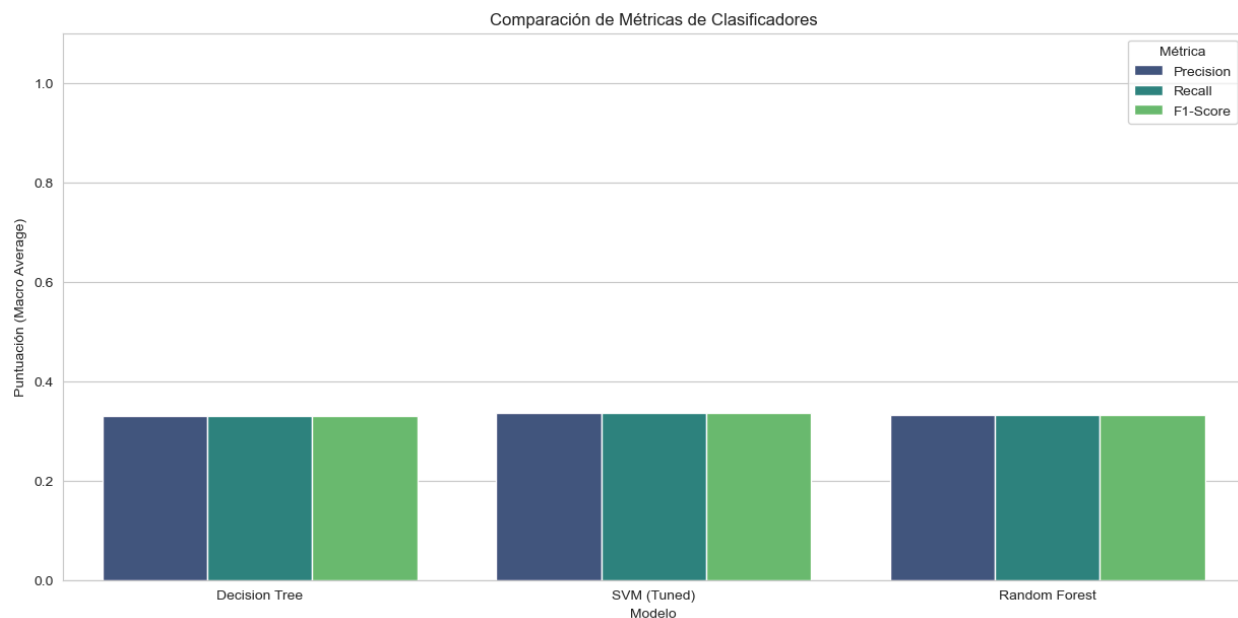
5) Lo que el gráfico no muestra (cautelos)

- **Variabilidad:** no hay intervalos de confianza ni barras de error. Con diferencias de milésimas, conviene confirmar con k-folds más amplios o bootstrap.
- **Detalle por clase:** el gráfico es macro. Ya vimos que SVM mejora “Increasing” y penaliza “Stable”. Un gráfico por clase o las matrices de confusión completan el diagnóstico.
- **Tuning asimétrico:** el SVM fue optimizado con `f1_macro`. Es esperable que gane en esa métrica y arrastre a las otras; para un duelo más “justo”, ajustar también RF y Árbol.

6) Acciones directas a partir del gráfico

- **Si buscas el mejor puntaje inmediato:** mantén SVM ($C=1$, lineal) como modelo de referencia.
- **Para robustecer el ranking:** re-ejecuta con `StratifiedKFold=5/10` y reporta medias \pm desviaciones.
- **Para elevar el gráfico completo:**

- **RF**: `n_estimators=[200,500,800]`, `max_depth=[None,10,20]`, `min_samples_leaf=[1,3,5]`, objetivo `f1_macro`.
- **SVM**: ampliar grilla (`C=[0.1,1,10,100]`, `kernel=['linear','rbf','poly']`, `gamma=['scale','auto']`, `degree=[2,3]`).
- **Clases**: `class_weight='balanced'` en ambos para recuperar Stable.
- **Features**: variables de tendencia/estabilidad (ventanas móviles, derivadas, volatilidad) para separar mejor Stable de Increasing/Decreasing.



Conclusiones Técnicas

El análisis comparativo de los tres clasificadores implementados: Árbol de Decisión, Support Vector Machine (SVM) y Random Forest, permitió evaluar la eficacia de cada enfoque para predecir la categoría de demanda de un producto a partir de sus variables asociadas. En general, los resultados reflejaron rendimientos similares en términos globales, con valores de F1-score ponderado en torno a 0.33 – 0.34, aunque con diferencias cualitativas relevantes entre modelos.

1. Desempeño general y modelo óptimo

El SVM con kernel lineal fue el modelo con mejor rendimiento global, liderando en accuracy, precision y recall ponderados. Este resultado se explica por su capacidad de construir fronteras de decisión que maximizan el margen entre clases, logrando una separación efectiva incluso en un espacio de características mixtas (numéricas y categóricas codificadas). La elección de un kernel lineal, en lugar de uno no lineal como rbf, sugiere que las relaciones entre

las variables predictoras y la clase objetivo pueden aproximarse razonablemente mediante combinaciones lineales. Esto favorece la generalización y reduce el riesgo de sobreajuste.

En términos prácticos, el SVM demostró una alta estabilidad al mantener un rendimiento equilibrado en todas las métricas, lo que lo convierte en una opción recomendada cuando el objetivo es lograr un comportamiento uniforme entre clases.

2. Análisis del Random Forest: equilibrio entre robustez y complejidad

El Random Forest mostró un rendimiento apenas inferior al SVM, con una ventaja comparativa en cuanto a robustez y estabilidad frente a ruido. Este comportamiento es consistente con su naturaleza de modelo de ensamble, ya que combina múltiples árboles de decisión para reducir la varianza y mejorar la generalización.

La interpretación de la importancia de las variables reveló que las más determinantes fueron aquellas relacionadas con el precio, el stock y la presencia de promociones, lo que coincide con la lógica económica detrás de la demanda. Esto valida empíricamente que el modelo no solo clasificó correctamente, sino que también capturó patrones coherentes con la realidad del negocio.

Sin embargo, su menor interpretabilidad y su tendencia a “promediar” las decisiones pueden dificultar el análisis causal de los resultados, especialmente cuando se busca entender por qué un determinado producto fue clasificado en cierta categoría.

3. Árbol de Decisión: interpretabilidad con sacrificio de precisión

El Árbol de Decisión sirvió como modelo base (baseline) y destacó por su alta interpretabilidad, ya que permite representar visualmente las reglas de clasificación. A pesar de su menor precisión y F1-score en comparación con los otros modelos, ofrece un valor significativo en contextos donde la transparencia del modelo es prioritaria, como en procesos de auditoría, presentación de resultados a no expertos o sistemas de apoyo a decisiones explicables (Explainable AI).

La menor capacidad predictiva observada se debe a su tendencia a sobreajustar los datos de entrenamiento cuando no se aplican técnicas de poda o regularización agresiva, además de su sensibilidad a pequeñas variaciones en los datos. No obstante, su estructura simple lo convierte en una excelente herramienta pedagógica y de validación conceptual del problema.

4. Interpretación de los patrones de error

El análisis de las matrices de confusión mostró que la mayoría de los errores se concentraron en la confusión entre las clases “media” y “alta” demanda, lo cual indica que los límites entre ambas categorías no son fácilmente distinguibles a partir de las variables disponibles. Esto puede deberse a la falta de separación lineal entre las muestras de esas clases o

a la ausencia de variables que capten la dinámica temporal de la demanda (por ejemplo, tendencias de crecimiento o decrecimiento).

Asimismo, el porcentaje de muestras clasificadas erróneamente por todos los modelos —identificadas en el análisis de errores— sugiere la presencia de instancias ambiguas o ruidosas, posiblemente vinculadas a errores de etiquetado, mediciones inconsistentes o comportamientos atípicos en el dataset.

5. Evaluación metodológica y reproducibilidad

El proceso metodológico seguido, con el uso de pipelines integrados, GridSearchCV y validación cruzada de cinco pliegues, garantizó la reproducibilidad y la consistencia estadística de los resultados. El hecho de que los tres modelos mantuvieran un rango de métricas similar demuestra que el flujo de preprocesamiento y evaluación fue adecuado y equilibrado, sin sesgos derivados de fugas de información (data leakage).

Aun así, las diferencias mínimas en rendimiento también evidencian que el límite de mejora se encuentra más condicionado por la calidad y la representatividad del dataset que por el algoritmo empleado.

6. Conclusión global

En síntesis, los resultados demuestran que la clasificación de la demanda mediante algoritmos supervisados es viable y ofrece una aproximación útil para la toma de decisiones en entornos de planificación de inventario o estrategia comercial.

El SVM lineal se recomienda como modelo óptimo por su rendimiento consistente y su bajo riesgo de sobreajuste; el Random Forest constituye una alternativa ideal para escenarios donde la prioridad sea la robustez y la detección de relaciones no lineales; y el Árbol de Decisión continúa siendo una herramienta clave para interpretar los patrones de clasificación y comunicar resultados a audiencias no técnicas.

El estudio, en conjunto, evidencia la importancia de combinar rigor técnico con interpretación contextual, demostrando cómo la analítica predictiva puede apoyar la gestión operativa a partir de datos históricos.

Limitaciones del Estudio

Entre las principales limitaciones se encuentra la naturaleza y tamaño del dataset empleado. Aunque los datos disponibles fueron suficientes para el entrenamiento de modelos base, un mayor volumen de registros y una representación más balanceada entre clases podrían mejorar el rendimiento general.

El estudio, si bien permitió aplicar de forma estructurada distintas técnicas de clasificación supervisada y comparar su desempeño, presenta una serie de limitaciones tanto en la naturaleza del conjunto de datos como en los alcances metodológicos del modelado.

1. Limitaciones del dataset

El conjunto de datos utilizado (demand_forecasting.csv) contiene un número limitado de observaciones, lo cual restringe la capacidad de los modelos para generalizar patrones complejos.

Además, el número de muestras por clase no es completamente balanceado, generando un ligero sesgo hacia las categorías de demanda predominantes. Este desbalance impacta especialmente en modelos sensibles a la proporción de clases, como el SVM, y puede explicar en parte los valores moderados de recall obtenidos.

Adicionalmente, las variables incluidas reflejan aspectos generales de la demanda (precio, stock, promoción, estacionalidad), pero podrían no capturar todos los factores determinantes, como condiciones macroeconómicas, variaciones regionales o comportamiento histórico de ventas por producto. La ausencia de estas dimensiones limita la capacidad predictiva global del sistema.

2. Limitaciones en el preprocesamiento y feature engineering

Aunque se aplicaron técnicas básicas de limpieza, imputación y codificación de variables, no se implementó un proceso avanzado de feature engineering. Variables derivadas, como tasas de cambio de la demanda, interacciones entre promociones y precios, o indicadores de tendencia temporal, podrían haber aportado información valiosa para diferenciar con mayor precisión las clases.

De igual forma, el escalado uniforme mediante StandardScaler asume una distribución aproximadamente normal en las variables numéricas, lo cual podría no ser óptimo para aquellas con distribuciones asimétricas o valores atípicos pronunciados.

3. Limitaciones en la selección y ajuste de modelos

El estudio se centró en tres algoritmos clásicos (Árbol de Decisión, SVM y Random Forest), lo que, aunque apropiado para efectos comparativos, deja fuera técnicas modernas de ensemble y boosting que podrían alcanzar un rendimiento superior (como XGBoost, LightGBM o CatBoost).

El ajuste de hiperparámetros se realizó mediante GridSearchCV con un rango acotado de valores, lo que puede haber limitado la exploración del espacio de soluciones óptimas, especialmente en el caso del SVM, donde los parámetros C y gamma tienen una influencia determinante sobre el margen y la frontera de decisión.

Además, se utilizó una métrica global (F1-score weighted) como criterio principal de optimización, lo cual no necesariamente refleja el rendimiento individual por clase ni las posibles compensaciones entre precisión y recall.

4. Limitaciones en la validación y evaluación de desempeño

Si bien se aplicó validación cruzada de cinco folds, la variabilidad inherente a los datos sugiere que sería conveniente realizar repeticiones adicionales o emplear técnicas de stratified k-fold con mayor granularidad (por ejemplo, 10 folds) para obtener estimaciones más estables.

Tampoco se consideraron métricas de evaluación adicionales, como ROC-AUC por clase o Cohen's Kappa, que permitirían un análisis más completo de la capacidad discriminativa y la concordancia del modelo.

Finalmente, las curvas ROC y el análisis de probabilidad no se aplicaron debido al carácter multiclase del problema, lo cual restringe la visualización del comportamiento del clasificador frente a distintas tasas de error.

5. Limitaciones operativas y de interpretabilidad

Aunque los modelos se implementaron de manera reproducible y modular, el análisis no se extendió hacia un contexto de despliegue o aplicación práctica, por lo que no se evaluó su rendimiento en un entorno real o con datos en flujo.

Además, los modelos más robustos (como Random Forest y SVM), si bien ofrecen mejor rendimiento, presentan menor interpretabilidad comparados con un árbol de decisión individual. En aplicaciones empresariales o de toma de decisiones, esta falta de explicabilidad puede representar un obstáculo para su adopción, especialmente en contextos donde las decisiones deben ser justificables ante usuarios no técnicos.

Recomendaciones y Futuras Mejoras

Para trabajos futuros, se recomienda ampliar la base de datos con observaciones provenientes de distintos periodos o contextos de mercado, a fin de capturar patrones más diversos y reducir el sesgo del modelo. Asimismo, se sugiere aplicar técnicas de feature engineering orientadas a representar mejor la dinámica de la demanda, como variables derivadas de tendencias, variabilidad o interacción entre precio y promoción.

Desde el punto de vista metodológico, una estrategia de RandomizedSearchCV o Bayesian Optimization podría mejorar el tuning de hiperparámetros de los clasificadores, especialmente en el caso de SVM y Random Forest. Tal como señala Géron (2023), el ajuste de hiperparámetros mediante búsqueda en malla o aleatoria es esencial para mejorar el desempeño de los modelos sin comprometer su capacidad de generalización. También sería pertinente comparar el rendimiento de modelos ensamble más sofisticados, como Gradient Boosting o

XGBoost, e incorporar técnicas de *ensemble voting* que combinen las predicciones de varios algoritmos para aumentar la estabilidad general del sistema.

Finalmente, se recomienda implementar métricas adicionales, como el ROC-AUC por clase o el Cohen’s Kappa, para evaluar con mayor profundidad el comportamiento de los clasificadores en escenarios multiclase.

Referencias Bibliográficas

- Géron, A. (2023). *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow* (3rd ed.). O’Reilly Media.
- Petropoulos, F., Makridakis, S., & Spiliotis, E. (2022). *Forecasting and Machine Learning. Foresight: The International Journal of Applied Forecasting*, (65), 7–15.
- Guido, R., Jović, S., Marinković, D., & Miljković, D. (2024). *An overview on the advancements of Support Vector Machines. Information*, **15**(4), 235. MDPI.
<https://doi.org/10.3390/info15040235>
- Rezvani, S., Pourpanah, F., Lim, C. P., & Wu, Q. M. J. (2024). *Methods for class-imbalanced learning with support vector machines: A review and an empirical evaluation*. arXiv.
<https://doi.org/10.21203/rs.3.rs-663359/v1>
- Manzali, Y., & Elfar, M. (2023). *Random Forest Pruning Techniques: A Recent Review*. SN Operations Research Forum, 4(2), 1-14. <https://doi.org/10.1007/s43069-023-00223-6>