Josue Paredes Méndez
00345132

Reporte Final - Predicción de Ventas e Inventario

1. Título y Objetivo del Proyecto

- **Título:** Predicción de Ventas e Inventario para Optimización del Stock.
- **Objetivo general:** Desarrollar un modelo predictivo para anticipar la cantidad de ventas por producto, con el fin de optimizar el manejo de inventario y reducir costos operativos por sobrestock o desabastecimientos.
- **Resumen ejecutivo:** Se buscó generar valor mediante un enfoque de ciencia de datos aplicado al negocio, mejorando la planificación de stock y reduciendo pérdidas por errores de aprovisionamiento. El modelo se entrena con datos históricos de un ERP, y proporciona predicciones útiles para la toma de decisiones semanales.
- **Visión técnica:** Se plantea como hipótesis que existen patrones temporales y de comportamiento de productos que permiten anticipar con precisión la cantidad de productos vendidos. Se plantea el uso de modelos supervisados para resolver este reto.

2. Contexto y Alcance

- **Situación actual:** La empresa cuenta con un ERP que registra cada transacción de venta, incluyendo fecha, producto y precio. Sin embargo, la planificación de inventario aún se realiza de forma empírica.
- **Dolor del negocio:** La variabilidad de la demanda genera frecuentes errores de stock (desabastecimiento y exceso).
- **Diagnóstico inicial:** Se analizaron datos históricos y entrevistas con el equipo para validar que la predicción de ventas es viable.
- **Alcance:** Se incluye la predicción de cantidad vendida por producto usando algoritmos de regresión, con enfoque temporal. No se incluyen modelos de clasificación ni estrategias de pricing.

3. Entendimiento de los Datos

Se utilizaron datos del sistema ERP y el dataset 'SAP Bikes Sales' de Kaggle. Se trata de información estructurada, transaccional e histórica.
Volumen: Más de 1,000 registros y 20+ productos distintos.
Problemas detectados: Presencia de valores nulos, fechas como strings, outliers en PRICE_UNITARIO, y alta cardinalidad en PRODUCTID.
Se realizaron visualizaciones como histogramas de cantidad, boxplots de precios y curvas de tendencia mensual.

4. Preparación de los Datos

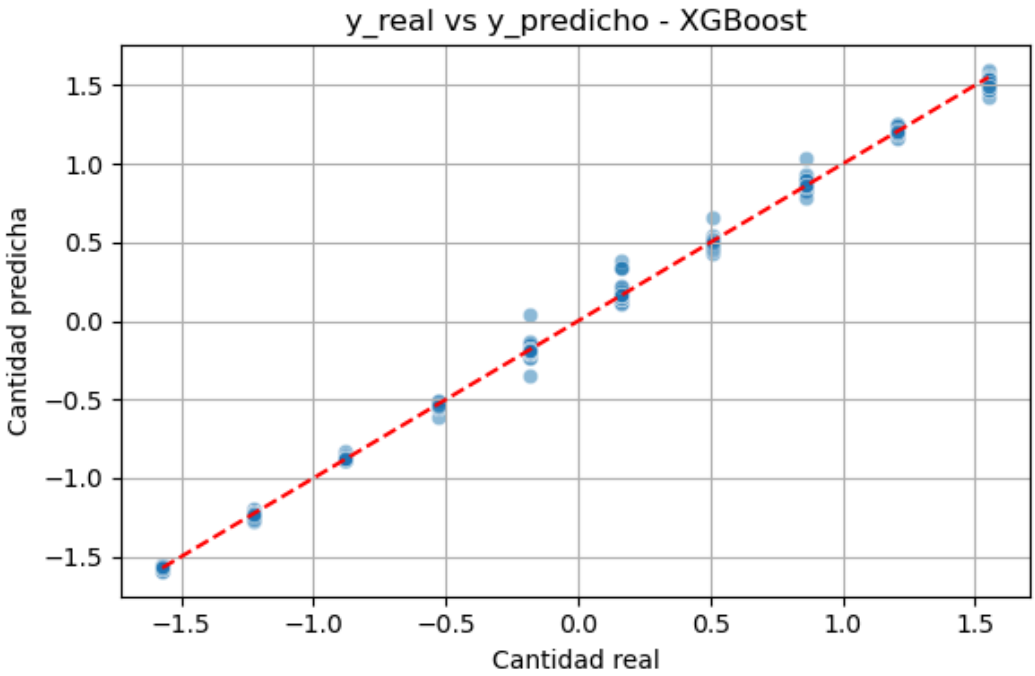
Limpieza: Se transformaron las fechas a formato datetime, se imputaron valores nulos de PRICE_UNITARIO, y se eliminaron columnas irrelevantes.
Integración: Se fusionaron los datos de ventas con información de productos.
Variables creadas: INGRESO_TOTAL (PRICE_UNITARIO * CANTIDAD), y variables temporales como MES, DÍA, y codificación de PRODUCTID.

5. Modelado

Modelos entrenados: Regresión Lineal, Árbol de Decisión, Random Forest, Gradient Boosting, y XGBoost.
Métricas utilizadas: MAE, RMSE y R^2.
Comparativa: XGBoost tuvo el mejor desempeño con RMSE de 0.033 y R^2 superior a 0.998.
Random Forest y Decision Tree también ofrecieron buenos resultados. Regresión Lineal fue el peor modelo.

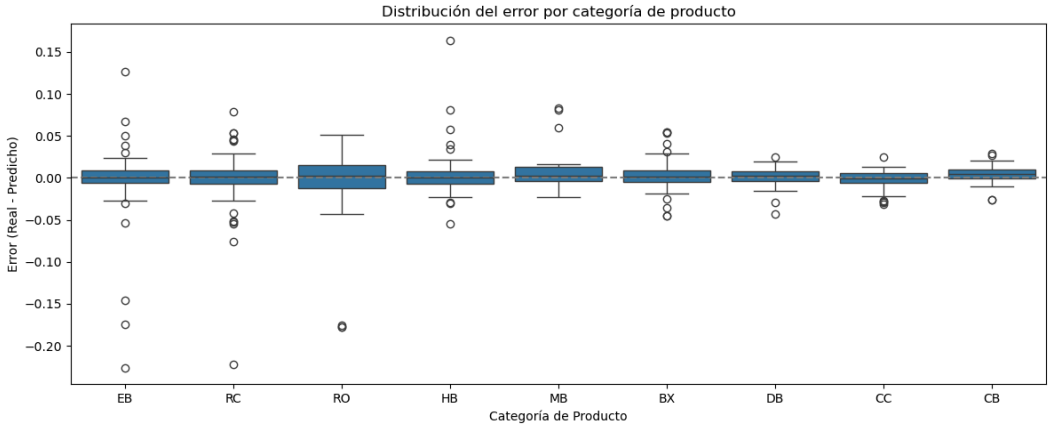
	MAE	RMSE	R2
XGBoost	0.016645	0.033851	0.998820
DecisionTree	0.005450	0.043484	0.998053
RandomForest	0.028148	0.058121	0.996521
GradientBoosting	0.090702	0.116929	0.985920
LinearRegression	0.548816	0.699963	0.495453

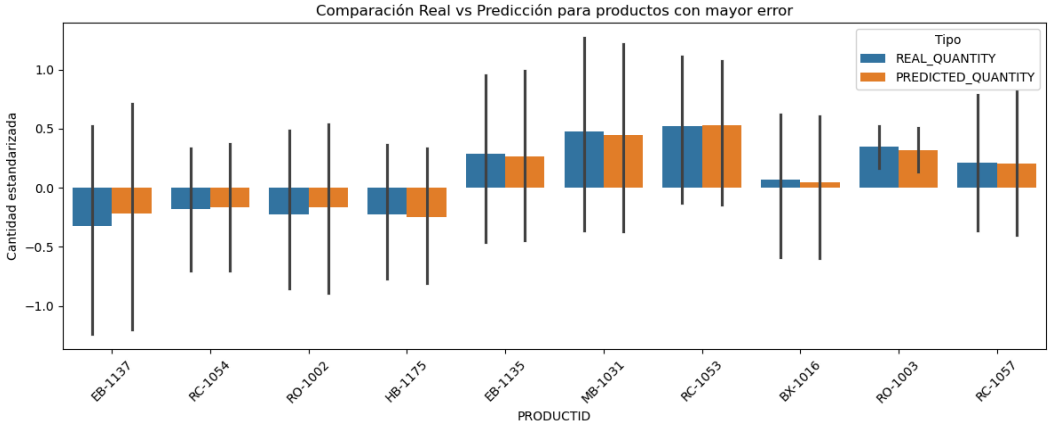
	R2 promedio	R2 std	RMSE promedio	RMSE std
XGBoost	0.996569	0.000764	0.058377	0.007544
RandomForest	0.992679	0.001894	0.083080	0.009904
DecisionTree	0.988163	0.000990	0.103218	0.015218
GradientBoosting	0.984053	0.001110	0.126748	0.005591
LinearRegression	0.509249	0.016492	0.702583	0.016735



6. Evaluación e Interpretación

- El modelo de árboles mostró alta precisión en general.
- Linear Regression falló por no modelar relaciones no lineales.
- XGBoost fue el más estable y preciso, especialmente validado con cross-validation.
- Se generaron gráficas de Feature Importance, PDP, SHAP y curva de aprendizaje para interpretar.





7. Plan de Implementación

- Se plantea su integración en el ERP mediante una API RESTful que reciba el estado actual de productos y devuelva predicciones de venta semanal.
- Monitoreo de métricas como RMSE y error absoluto medio.
- Se recomienda reentrenar el modelo una vez al mes.

8. Conclusiones y Recomendaciones

- El modelo XGBoost predice la variable CANTIDAD con alta precisión.
- INGRESO_TOTAL y PRICE_UNITARIO son las variables más relevantes.
- El modelo es adecuado para implementación operativa.
- Se recomienda añadir variables externas (marketing, clima) para futuras mejoras.

9. Apéndices

Scripts utilizados:
- modeling.py
- feature_engineering.py

Visualizaciones:
- Importancia de variables
- Comparaciones reales vs predicho
- Boxplots por categoría

Librerías: pandas, scikit-learn, xgboost, matplotlib, seaborn, numpy, sklearn, shap