****

**Predicción de Ventas con Machine Learning**

****

Machine Learning I

****

Elaborado por:

**Inuma Macedo Mariana Milagros**

Solicitado por:

**Sergio Victor Orizano Salvador**

Huancayo, 2025

# Introducción

El presente informe documenta el desarrollo completo de un proyecto de machine learning orientado a predecir el comportamiento de compra de los clientes en una tienda virtual de productos electrónicos. El objetivo principal fue construir modelos predictivos que permitan estimar tanto si un cliente realizará una compra como el monto que gastará en caso de hacerlo.

El proyecto abarcó desde la generación de datos sintéticos hasta la evaluación de modelos de regresión y clasificación, incluyendo visualización, validación cruzada, y almacenamiento de resultados.

# **Generación de Datos Sintéticos**

Dado que no se disponía de datos reales, se optó por generar un conjunto de datos completamente sintético pero realista, basado en el siguiente prompt:

“Genera un conjunto de datos sintéticos para una tienda virtual de productos electrónicos que desea analizar el comportamiento de sus clientes. Los datos deben parecer realistas pero deben ser completamente ficticios.

Genera 15,000 registros y guarda el resultado en un archivo CSV llamado clientes\_tienda\_virtual.csv.

A partir de este requerimiento, se desarrolló el script generar\_datos\_clientes.py, que produce un archivo CSV con 15,000 registros bajo la estructura:

* Cliente\_ID: código alfanumérico único, formato CL-00001, CL-00002, …
* grupo\_edad: categoría (18-25, 26-35, 36-45, 46-60, 60+)
* region: zona geográfica (Norte, Centro, Sur, Oriente)
* nivel\_ingresos: rango de ingresos mensuales (bajo, medio, alto)
* horas\_conectado: tiempo promedio semanal navegando en la tienda (0–20 horas)
* clics\_en\_productos: número de productos sobre los que hizo clic (0–50)
* uso\_cupones: cantidad de cupones promocionales usados (0–5)
* compra\_realizada: 0 = no realizó compra, 1 = sí compró
* total\_compra: monto total en soles de la compra (0 si no compró, entre 100 y 4000 si compró)

La variable compra\_realizada debe depender de variables como clics\_en\_productos, uso\_cupones, y nivel\_ingresos, mientras que total\_compra debe ser mayor si el cliente compró y tiene ingresos altos.

Los datos deben ser consistentes, sin valores faltantes, y exportados como archivo CSV”.

La lógica de compra y monto fue diseñada para depender de factores como el uso de cupones, nivel de ingresos y actividad dentro de la tienda. El archivo generado clientes\_tienda\_virtual.csv se almacena en la carpeta data\_sintetica.

# ****Exploración y Preparación****

En el **notebook** **01\_exploracion\_clientes.ipynb** se analizaron las características de la base de datos:

* Se verificó la ausencia de valores nulos.
* Se examinó la distribución de cada variable.
* Se observó la influencia del nivel de ingresos y el uso de cupones en la probabilidad de compra.
* Se descartaron los registros con total\_compra = 0 para los modelos de regresión, ya que estos clientes no realizaron compra.

Las variables categóricas fueron transformadas con OneHotEncoder, mientras que las variables numéricas se procesaron directamente o se transformaron polinómicamente en el modelo correspondiente.

# ****Modelos Utilizados****

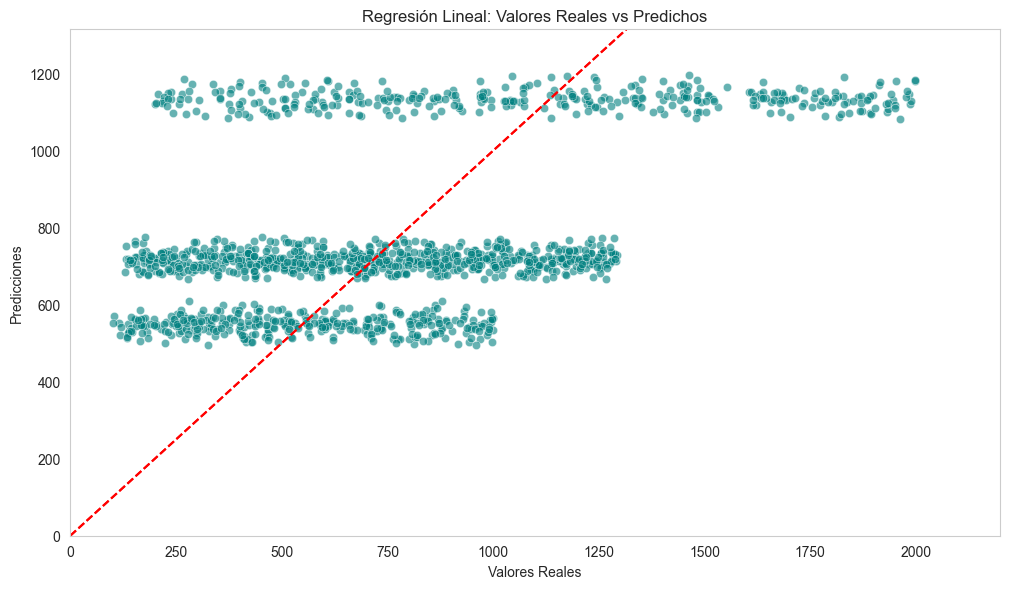
### ****4.1 Regresión Lineal****

**Objetivo:** Predecir el monto total de compra (total\_compra), considerando solo a clientes que realizaron una compra.

**Modelo:** Regresión lineal simple con pipeline de preprocesamiento.

**Evaluación:**

* RMSE (Test): 373.30
* R² (Test): 0.23·
* R² promedio (CV): 0.24

**Gráfico 1: Dispersión real vs predicción – Regresión Lineal**

Se observa una gran dispersión de puntos, lo que refleja que muchos valores predichos están alejados de los valores reales. Esto evidencia la limitación del modelo para capturar relaciones complejas.

### ****4.2 Regresión Polinómica****

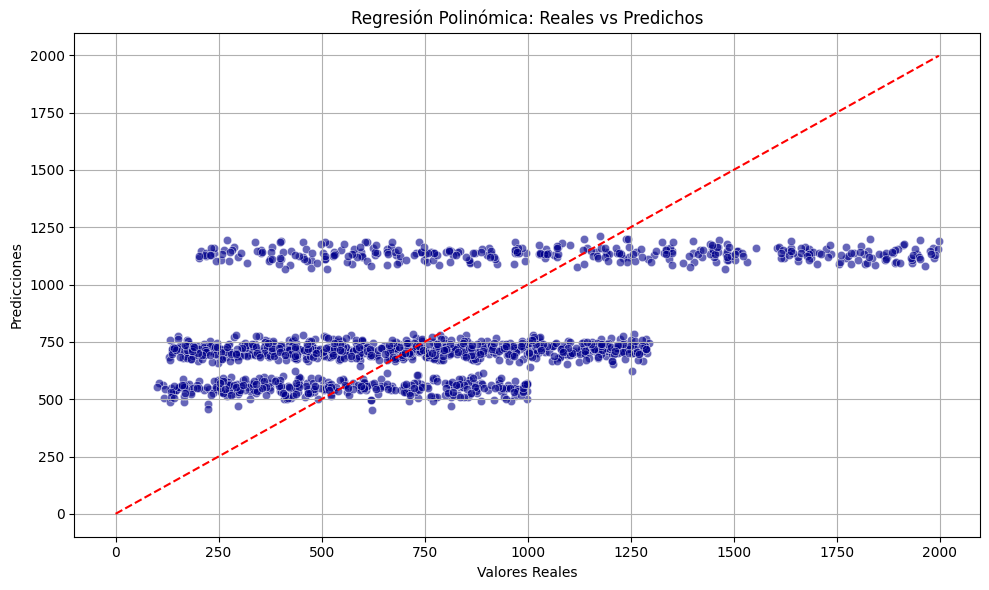
**Objetivo:** Mejorar la capacidad predictiva del modelo al incluir interacciones no lineales entre variables numéricas.

**Modelo:** Se empleó un pipeline con PolynomialFeatures de grado 2 y regresión lineal.

**Evaluación:**

* RMSE (Test): 373.26
* R² (Test): 0.24
* R² promedio (CV): 0.24

**Gráfico 2: Dispersión real vs predicción – Regresión Polinómica**



Este gráfico muestra qué tan cerca están las predicciones del modelo polinómico respecto a los valores reales. Como muchos puntos están lejos de la línea roja (ideal), el modelo no predice con mucha precisión y tiende a agrupar los resultados.

### ****4.3 Clasificación con Árbol de Decisión****

**Objetivo:** Predecir si un cliente realizará una compra (compra\_realizada = 1) o no (0).

**Modelo:** Árbol de decisión con profundidad máxima de 5. Se eligió este modelo por su alta interpretabilidad, capacidad de manejar datos categóricos y numéricos, y facilidad de visualización.

**Evaluación:**

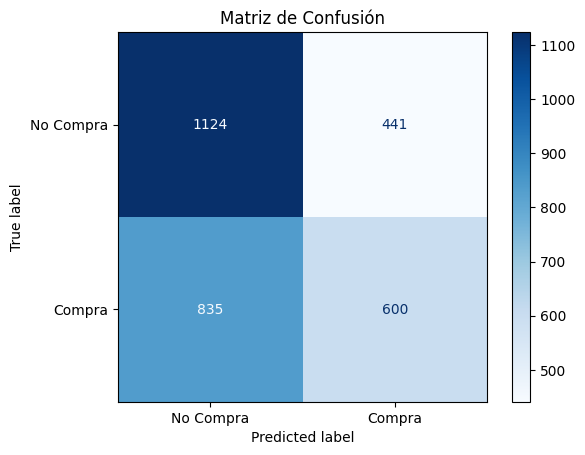
* Accuracy promedio (CV): 0.75

**Reporte de clasificación:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Clase** | **Precisión** | **Recall** | **F1-score** |
| **No compra** | 0.57 | 0.72 | 0.64 |
| **Compra** | 0.58 | 0.42 | 0.48 |

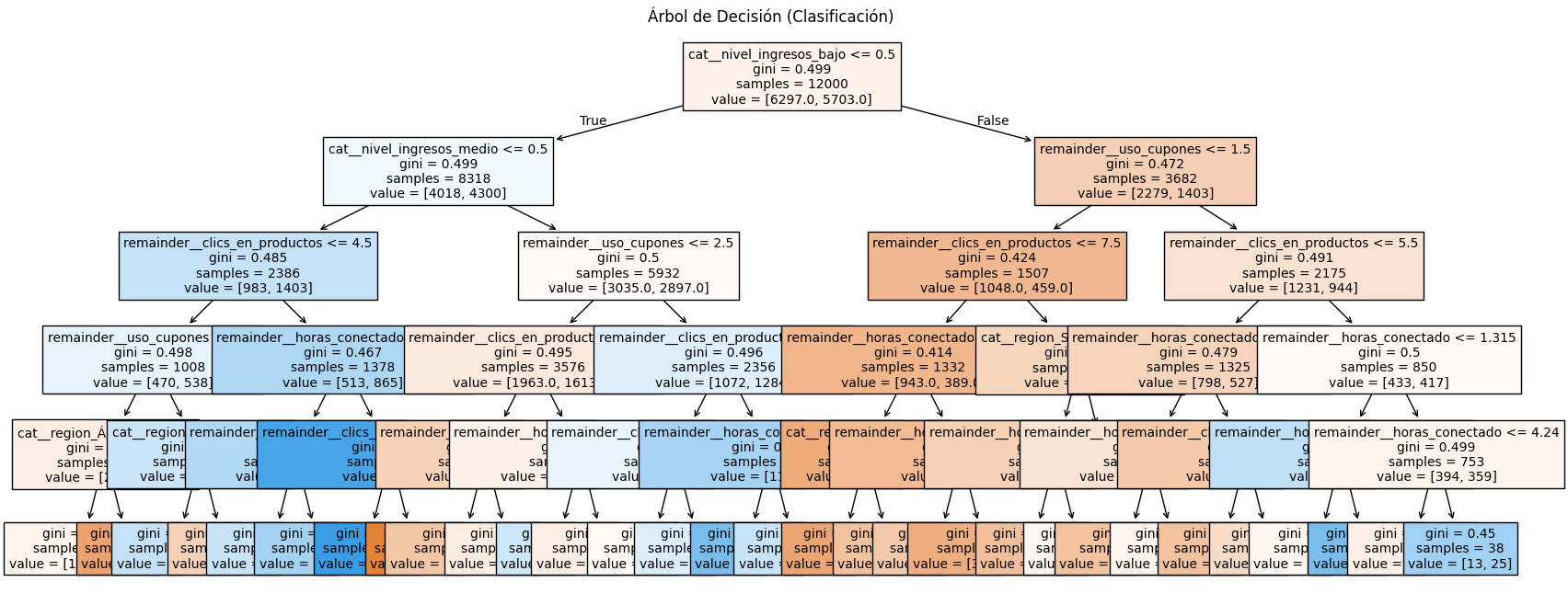
El modelo identifica mejor a los clientes que no compran (recall: 72%), pero tiene baja capacidad para detectar a quienes sí compran (recall: 42%). Esto indica un rendimiento desequilibrado, por lo que se recomienda considerar ajustes al árbol o probar modelos más robustos como Random Forest o XGBoost.

**Gráfico 3: Matriz de confusión**



**Esta matriz de confusión muestra el desempeño del modelo de clasificación al predecir si un cliente realizará una compra. El modelo acertó en 1124 casos en los que no se hizo una compra y en 600 casos en los que sí se compró. Sin embargo, se equivocó al predecir que 835 clientes no comprarían cuando sí lo hicieron, y que 441 comprarían cuando en realidad no lo hicieron. Esto indica que el modelo tiene mejor desempeño detectando a quienes no compran, pero le cuesta más identificar correctamente a quienes sí realizan una compra.**

**Gráfico 4: Visualización del árbol de decisión**



Este gráfico muestra cómo el modelo de árbol de decisión clasifica a los clientes según sus características, como nivel de ingresos, clics en productos y uso de cupones. Cada división representa una decisión basada en una variable, y los colores indican si el cliente probablemente comprará (naranja) o no (azul). Es útil para entender qué factores influyen más en la decisión de compra.

# ****Visualización de Resultados****

Los siguientes gráficos fueron generados para facilitar la interpretación de los modelos:

* **Gráfico 1:** Dispersión entre valores reales y predichos para regresión lineal.
* **Gráfico 2:** Dispersión ajustada para regresión polinómica.
* **Gráfico 3:** Matriz de confusión para modelo de clasificación.
* **Gráfico 4:** Árbol de decisión entrenado y visualizado.

# ****Exportación de Resultados****

Los modelos y sus resultados fueron almacenados para su reutilización o análisis posterior:

* **Modelos entrenados (**.pkl**)** se guardan en la carpeta modelos\_guardados/
* **Predicciones (**.csv**)** y conjuntos de prueba (X\_test, y\_test) están disponibles en resultados\_modelos/, separados por regresión y clasificación.

Esta estructura permite compartir fácilmente los resultados con otros analistas o integrarlos a una aplicación.

# ****Conclusiones****

Este proyecto permitió aplicar técnicas de machine learning en un flujo de trabajo completo: desde la generación de datos hasta la evaluación de modelos. Entre los hallazgos clave se destacan:

* La regresión polinómica ofrece una leve mejora frente a la regresión lineal, pero no logra capturar del todo la complejidad del gasto del cliente.
* El modelo de clasificación con árboles de decisión resultó ser el más robusto, combinando buen rendimiento con interpretabilidad.
* La calidad de los datos influye notablemente en los resultados. Aun siendo sintéticos, la lógica interna permitió simular patrones realistas.

# **Recomendaciones**

* **Obtener datos reales:** Esto permitiría entrenar modelos con comportamientos más fieles al entorno de negocio.
* **Optimización de hiperparámetros:** Puede mejorar el rendimiento del árbol de decisión y otros modelos más complejos.
* **Ampliar modelos de clasificación:** Probar algoritmos como Random Forest o XGBoost para mejorar precisión.
* **Evaluar importancia de variables:** Incluir herramientas como SHAP o Permutation Importance.

# **Repositorio GitHub**

* https://github.com/Mar28M/Prediccion\_ventas\_ML1\_IDL1.git