

Máster en Data Science

---

# De la predicción a la decisión: Simulador de descuentos para maximizar margen

**Autores:**

Luis Carlos Deza Monge

Álvaro Hernández Rubio

**Tutor:**

Antonio Pita Lozano

**Fecha:**

14 de Febrero de 2026

# Índice

<b>Objetivo del TFM</b>	<b>5</b>
<b>Preguntas de investigación</b>	<b>6</b>
<b>Alcance y exclusiones</b>	<b>6</b>
<b>Resumen</b>	<b>7</b>
<b>1. Introducción</b>	<b>8</b>
<b>2. Diseño e Implementación</b>	<b>9</b>
<b>2.1. Ficheros utilizados en el pipeline</b>	<b>9</b>
<b>2.2. Granularidad de modelado y motivación</b>	<b>9</b>
<b>2.3 Exploración Inicial</b>	<b>9</b>
<b>2.4 Limpieza de datos</b>	<b>11</b>
<b>2.5. Construcción de variables de precio y descuento</b>	<b>12</b>
<b>2.6 Filtros de calidad</b>	<b>13</b>
<b>2.7 Productos Seleccionados</b>	<b>13</b>
<b>3. Feature Engineering</b>	<b>16</b>
<b>3.1 Variables de dinámica temporal (LAGs)</b>	<b>16</b>
<b>3.2 Variables de estacionalidad</b>	<b>16</b>
<b>3.3 Variables de contexto (priors de nivel)</b>	<b>17</b>
<b>3.4 Variable objetivo: Lift incremental</b>	<b>17</b>
<b>3.5 Resumen del dataset final</b>	<b>18</b>
<b>4. Modelo A (Baseline): especificación, supuestos e interpretación</b>	<b>19</b>
<b>4.1. Submuestra de entrenamiento: semanas sin descuento y corte temporal</b>	
	<b>19</b>
<b>4.2. Variables explicativas y su rol</b>	<b>19</b>
<b>4.3. Tratamiento de NaN e infinitos</b>	<b>19</b>
<b>4.4. Ajuste del modelo OLS y lectura del summary</b>	<b>20</b>
<b>4.5. Función predecir_baseline_ols y recorte a no negativo</b>	<b>20</b>
<b>4.6 Ajuste del Modelo A y validación del baseline</b>	<b>20</b>

<b>4.7. Riesgos y limitaciones del baseline</b>	20
<b>5. Modelo B (Lift): aprendizaje no lineal del efecto promocional</b>	22
<b>5.1 Variables promocionales y de intensidad: DISC_PCT</b>	22
<b>5.2 Interacción implícita con el nivel base: BASELINE_PRED</b>	22
<b>5.3 Dinámica reciente y memoria de la serie: lags LAG_1, LAG_2, LAG_4, LAG_12</b>	23
<b>5.4 Calendario y estacionalidad: WEEK_SIN, WEEK_COS y WEEK_OF_MONTH</b>	23
<b>5.5 Priors de producto y tienda: PROD_MEAN_SALES y STORE_MEAN_SALES</b>	23
<b>5.6 Resultado: un set de features “explicativo” del lift</b>	24
<b>5.7. Modelos candidatos y justificación</b>	24
<b>5.8. Hiperparámetros y trade-offs</b>	24
<b>5.9. Evaluación: MAE en ventas totales</b>	24
<b>5.10 Consideraciones sobre sesgo del baseline</b>	25
<b>5.11 Robustez y generalización</b>	25
<b>6. Simulador de escenarios promocionales: diseño, uso y limitaciones</b>	26
<b>6.1 Cómo interpretar el resultado</b>	27
<b>6.2 Agregación “TODAS”: cómo se obtiene y por qué se restringe</b>	28
<b>6.3 Cálculo del margen en el simulador</b>	28
<b>6.4 Limitaciones y precauciones de uso</b>	31
<b>7. Validación, diagnóstico y sistema de recomendación</b>	33
<b>7.1. Backtesting (CHECK): Real vs Pred</b>	33
<b>7.2. Diagnóstico de monotonía</b>	33
<b>7.3. Productos problemáticos y causas típicas</b>	33
<b>7.4. Score de confianza (0–1): componentes</b>	33
<b>7.5. Ranking final y decisión</b>	34
<b>7.6. Cómo usar el ranking en un caso real</b>	34
<b>8. Resultados, conclusiones, limitaciones y trabajo futuro</b>	35
<b>8.1. Resultados: qué se obtiene al ejecutar el pipeline</b>	35
<b>8.2. Principales conclusiones</b>	35

<b>8.3. Limitaciones</b>	35
<b>8.4. Trabajo futuro</b>	35
<b>Anexos</b>	<b>36</b>
Anexo A. Diccionario ampliado de variables	36

# Objetivo del TFM

El objetivo de este TFM es desarrollar un modelo de predicción de ventas capaz de capturar el efecto incremental de las promociones de precio (retail discount) en unidades vendidas, separándolo del componente baseline (demanda sin promoción). Adicionalmente, el trabajo transforma el modelo en una herramienta de simulación que no solo estima ventas, sino que permite comparar escenarios de descuento en términos de rentabilidad, ayudando a seleccionar el porcentaje de promoción más conveniente para maximizar margen.

# Preguntas de investigación

- ¿Cómo estimar un baseline de ventas sin promoción usando únicamente semanas no promocionadas?
- ¿Cómo modelar el lift promocional y su relación con la profundidad del descuento?
- ¿Cómo evaluar coherencia del modelo (monotonía) y robustez (backtesting)?
- ¿Cómo convertir el modelo en una herramienta utilizable por negocio (simulador y ranking)?

## Alcance y exclusiones

El alcance se centra en el descuento retail observado (RETAIL\_DISC) y en el impacto sobre unidades vendidas. El código carga tablas relacionadas con cupones y campañas, pero el modelo final no incorpora explícitamente el efecto de cupones, display o Mailer, ya que estos descuentos no los aplica la propia compañía, si no los proveedores. Asimismo, no se calcula margen real por falta de costes en la fuente; se imputa un coste de venta ficticio del 70% sobre el precio (sin descuentos).

# Resumen

Este Trabajo Fin de Máster desarrolla una solución de analítica avanzada para cuantificar y simular el efecto de promociones de precio en retail, utilizando datos transaccionales. El núcleo metodológico es una arquitectura de dos etapas que separa el componente baseline (demanda esperada sin promoción) del componente lift (incremento atribuible al descuento). La primera etapa estima con una regresión OLS entrenada únicamente en semanas sin promoción. La segunda etapa aprende el lift con modelos de boosting (XGBoost y CatBoost), incorporando interacciones no lineales entre descuento, contexto temporal y nivel de demanda.

El trabajo incluye además un simulador de escenarios promocionales que permite variar el porcentaje de descuento para una combinación producto–tienda–semana y obtener ventas predichas, y por ende margen en euros y en porcentaje. Así como módulos de validación (backtesting, checks de monotonía) y un sistema de ranking de productos que pondera uplift por un score de confianza. La principal limitación encontrada es la alta intermitencia de demanda en una parte importante del surtido, lo que reduce el número de series con señal suficiente; aun así, la propuesta resulta útil para productos con histórico mínimo y cierta continuidad. También encontramos la dificultad de una gran cantidad de surtido con pocas unidades de venta por semana, lo cual hace que la variación con promoción o sin ella sea prácticamente nula.

# 1. Introducción

La promoción de precio es una herramienta fundamental en la estrategia comercial de gran consumo: incrementa volumen a corto plazo, mueve inventario y mejora visibilidad. Sin embargo, también puede erosionar margen, provocar compras adelantadas y distorsionar la lectura de la demanda ‘normal’. Por ello, disponer de métodos que separen ventas base y ventas incrementales es clave en la planificación promocional.

En un escenario real, el analista se enfrenta a datos fragmentados (múltiples tablas), ruido transaccional, semanas con ausencia de ventas y promociones de distinta intensidad. Este TFM aborda ese reto con un pipeline reproducible: transforma transacciones a un panel semanal coherente, crea variables explicativas, entrena modelos y expone un simulador para comparar escenarios.

## 2. Diseño e Implementación

Se utiliza el dataset ‘The Complete Journey’ (dunnhumby), ampliamente empleado en ejercicios académicos para análisis de marketing y retail. La fuente se distribuye en múltiples ficheros CSV: transacciones, catálogo de productos, campañas, cupones y variables causales (exposición).

### 2.1. Ficheros utilizados en el pipeline

El código carga los siguientes ficheros (entre otros):

- transaction\_data.csv: transacciones con WEEK\_NO, STORE\_ID, PRODUCT\_ID, QUANTITY, SALES\_VALUE, RETAIL\_DISC.
- product.csv: atributos por producto (DEPARTMENT, COMMODITY\_DESC, SUB\_COMMODITY\_DESC, BRAND, etc.).
- causal\_data.csv: variables de exposición (display, mailer) para posibles extensiones.
- coupon.csv y coupon\_redempt.csv: información de cupones (no incorporada al modelo final).
- campaign\_desc.csv y campaign\_table.csv: metadatos de campañas (no incorporados al modelo final).

Utilizamos principalmente la tabla de transacciones donde ya se incluyen los descuentos que soporta la tienda, y por tanto la compañía a la que va destinada este simulador. También utilizamos la tabla producto como maestro de artículos.

### 2.2. Granularidad de modelado y motivación

Las transacciones a nivel de ticket contienen alto ruido y sparse events. Para que el aprendizaje sea más estable, se agrega por semana y por pareja producto–tienda. La semana es además una unidad natural para promociones (muchas campañas se planifican semanalmente) y para construir variables como lags o estacionalidad.

La variable objetivo-primaria es QUANTITY (unidades vendidas) agregada por semana. La variable promocional observable es RETAIL\_DISC, interpretada como descuento de precio visible (importe). Para facilitar comparaciones, se convierte en porcentaje de descuento (DISC\_PCT) a partir de PRICE y PRICE\_BASE.

### 2.3 Exploración Inicial

La Figura 2.1 muestra la distribución de ventas promedio semanales por producto tras eliminar un outlier estructural cuya escala distorsionaba el conjunto.

El análisis revela una fuerte concentración en niveles de venta reducidos. La mediana se sitúa en torno a 2,5 unidades por semana, mientras que la mayoría de los productos no superan las 3 unidades semanales en promedio. Esta evidencia confirma la presencia de demanda de baja rotación, característica habitual en surtidos amplios del retail.

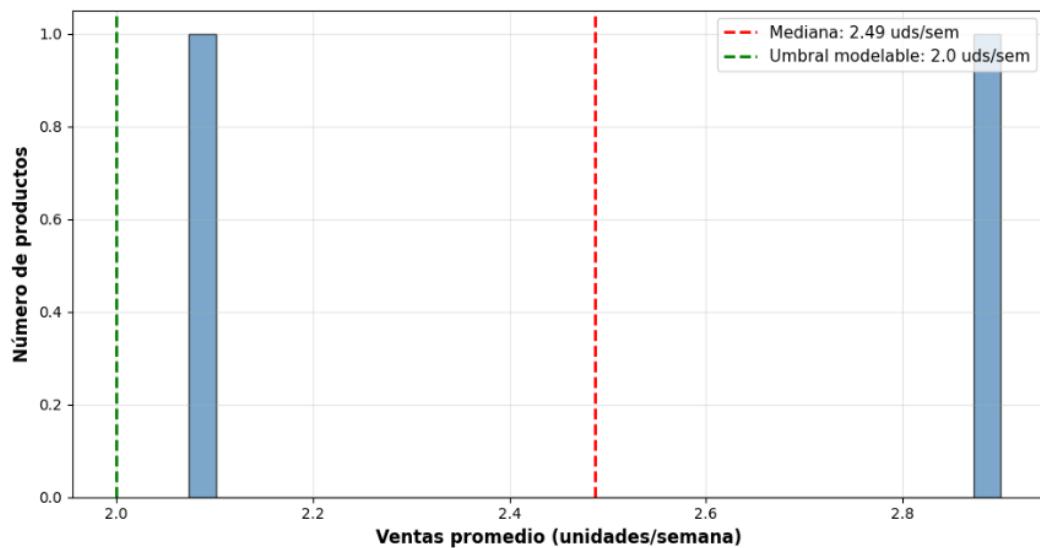


Figura 2.1. Distribución de ventas promedio por producto (99 productos tras eliminar outlier)

La Figura 2.2 presenta la distribución de los precios efectivos observados en el conjunto de datos, considerando las transacciones de los productos seleccionados para el análisis.

La mediana del precio se sitúa en torno a 1,00 dólar, lo que indica una fuerte concentración de observaciones en ese nivel. La distribución muestra una ligera asimetría positiva, con una cola hacia valores superiores, aunque la mayoría de los precios se agrupan en el rango aproximado de 0,80 a 1,50 dólares.

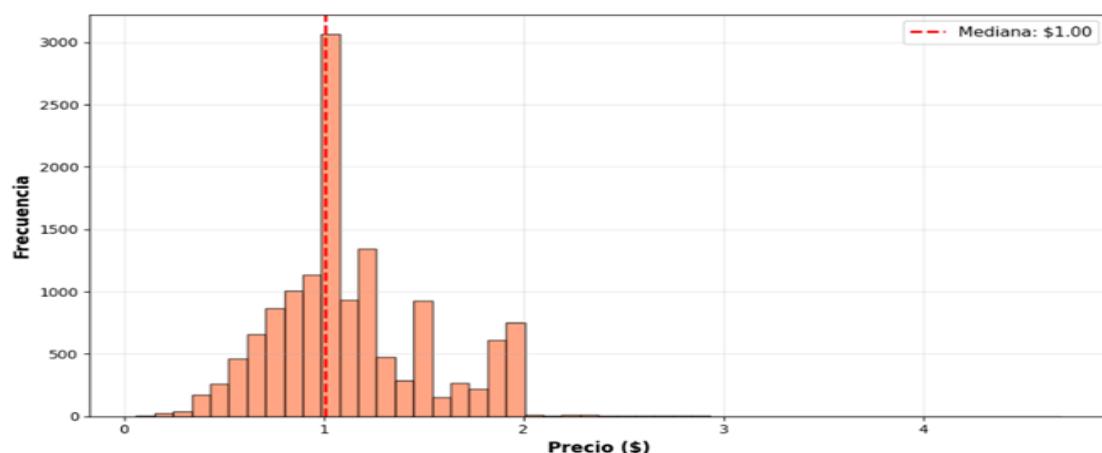


Figura 2.2. Distribución de precios efectivos.

La Figura 2.3 muestra la evolución del volumen de ventas según el nivel de descuento aplicado. Se observa una relación positiva entre la profundidad del descuento y las unidades vendidas: las semanas sin promoción presentan menores niveles de venta, mientras que los descuentos más elevados, especialmente superiores al 30%, generan mayores medianas y una mayor dispersión.

La presencia de valores extremos en los rangos de descuento altos refleja picos promocionales reales y no ruido estadístico. En conjunto, la evidencia confirma la existencia de efecto promocional y justifica la estimación separada del baseline y del lift incremental en el modelo propuesto.

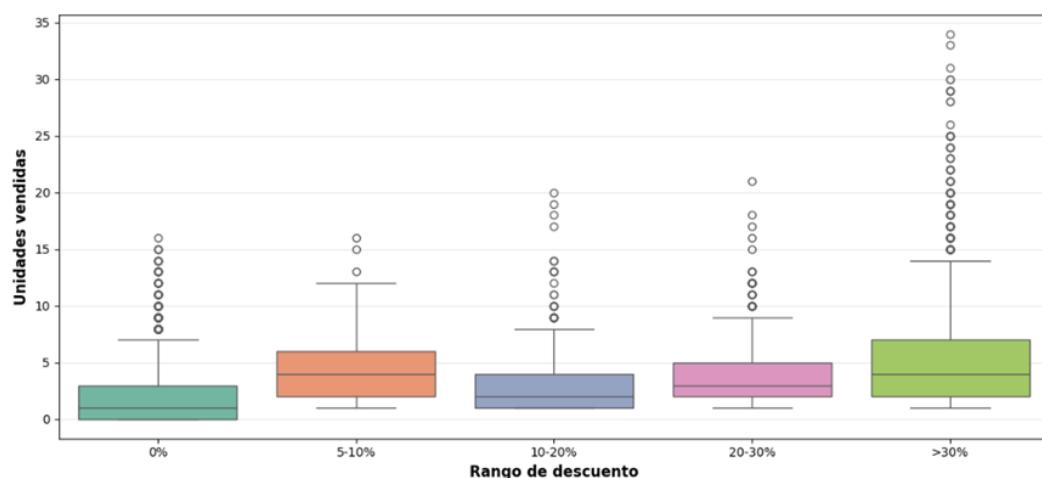


Figura 2.3. Distribución de ventas por profundidad de descuento.

## 2.4 Limpieza de datos

Esta sección describe, con detalle, cómo se construye el dataset final ‘wk’ (panel semanal producto–tienda) a partir de las tablas crudas. Cada decisión de filtrado busca equilibrar, mantener representatividad y garantizar señal suficiente para modelado.

Se elimina el tramo inicial del dataset, comenzando desde la semana 16. Este corte reduce la influencia de períodos potencialmente inestables (arranque, cambios de cobertura). Se analiza que la venta de las primeras 15 semanas no es comparable al resto de semanas, para no distorsionar el análisis las eliminamos. Según un enfoque de negocio, entendemos que son semanas de arranque.

La figura 2.4 muestra la evolución de las ventas totales semanales agregadas (en unidades) a lo largo del periodo disponible.

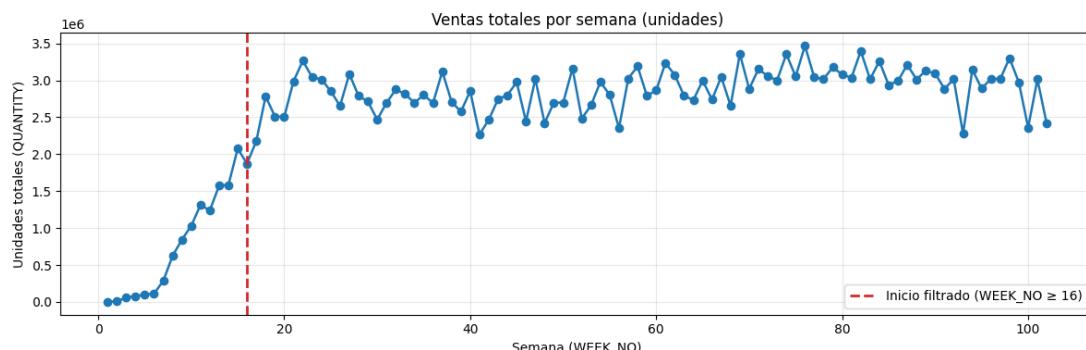


Figura 2.4. Ventas totales por semana (unidades)

## 2.5. Construcción de variables de precio y descuento

A partir de las variables transaccionales brutas se derivaron tres variables clave para el análisis: precio efectivo, precio base y porcentaje de descuento.

- **Precio efectivo (PRICE):** Representa el precio promedio pagado por unidad en cada combinación producto–tienda–semana. En ausencia de promoción aproxima el precio regular; en presencia de descuento refleja el precio efectivamente pagado.
- **Precio base (PRICE\_BASE):** Esta variable permite estimar el precio sin promoción y resulta esencial para calcular correctamente la intensidad relativa del descuento.
- **Porcentaje de descuento (DISC\_PCT):** El valor se acotó en el intervalo [0,0.9] para evitar distorsiones derivadas de errores de captura o promociones extremas no representativas.

El uso de descuento relativo (DISC\_PCT) en lugar del importe absoluto (RETAIL\_DISC) permite normalizar la intensidad promocional entre productos de distinto nivel de precio. Por ejemplo, un descuento de 0,50 dólares representa un 50% para un producto de 1 dólar, pero solo un 5% para uno de 10 dólares.

### Tratamiento de casos especiales

- Observaciones con QUANTITY=0 generan precios indefinidos y se excluyen antes del modelado.
- Casos excepcionales con descuentos negativos se reemplazan por 0.
- Valores de descuento superiores al 90% se recortan para evitar efectos extremos no estructurales.

## 2.6 Filtros de calidad

Dado el carácter altamente intermitente de la demanda, se aplican filtros secuenciales con el objetivo de garantizar robustez econométrica y evitar estimaciones inestables:

1. **Nivel mínimo de ventas:** ventas promedio  $\geq 2$  unidades/semana/tienda.
2. **Presencia temporal mínima:** ventas en al menos el 40% de las semanas.
3. **Exclusión de productos siempre en promoción.**
4. **Exigencia de al menos una semana con descuento.**

Los dos últimos criterios garantizan que exista variación suficiente en la variable promocional para identificar correctamente el baseline de demanda y el efecto incremental de las promociones.

Tras aplicar estos filtros, el surtido se reduce de 99 a 2 productos modelables. Esta reducción no se interpreta como una limitación metodológica, sino como un hallazgo empírico que evidencia la elevada intermitencia estructural del dataset y la dificultad de estimar elasticidades robustas para la mayoría de los SKUs en entornos de retail.

## 2.7 Productos Seleccionados

Tras la aplicación secuencial de los filtros de calidad, el dataset final quedó compuesto por 2 productos modelables, únicos que cumplen simultáneamente los criterios de volumen mínimo y presencia temporal suficiente.

### Producto 1082185

- Ventas promedio: 2,90 unidades/semana
- Presencia temporal: 82,8% de semanas con venta
- Frecuencia promocional: ~24%
- Precio base promedio: ~\$2,15

Este producto presenta demanda relativamente estable y alta continuidad temporal. Su elevada presencia semanal permite estimar patrones dinámicos (lags y estacionalidad) con mayor robustez. Asimismo, muestra respuesta positiva a promociones, lo que lo convierte en el caso más adecuado para modelado de elasticidad y análisis de lift incremental.

Desde una perspectiva de negocio, puede considerarse un producto de rotación consolidada dentro del surtido analizado.

## Producto 995242

- Ventas promedio: 2,07 unidades/semana
- Presencia temporal: 55,7% de semanas con venta
- Frecuencia promocional: ~19%
- Precio base promedio: ~\$3,42

Este producto presenta mayor intermitencia relativa y mayor volatilidad semanal. Aunque cumple los criterios mínimos de modelabilidad, su menor continuidad temporal implica estimaciones potencialmente menos estables que en el caso anterior.

### 2.8. Construcción de un panel regular (grid completo)

En datos transaccionales, la ausencia de registros para una combinación producto–tienda–semana puede deberse a múltiples causas, como roturas de stock, salida temporal del surtido, errores de integración o una ausencia genuina de demanda. En este trabajo se adopta la hipótesis operativa estándar en paneles de retail agregados: cuando no se registra ninguna transacción en una semana, se asume que las ventas son cero. Esta decisión se justifica por la ausencia de una variable fiable de stock o disponibilidad en el dataset y por la necesidad de garantizar coherencia temporal en el modelado predictivo.

Al agregar transacciones a nivel semanal, el resultado natural es una serie irregular con semanas ausentes. Si estos huecos no se rellenan, variables de rezago como LAG\_1 dejarían de representar “la semana anterior” y pasarían a referirse a la observación previa disponible, que podría estar separada por varias semanas. Esto rompe el significado temporal de los retardos y degrada la calidad del modelo.

Para evitarlo, se construye un panel regular o grid completo, generando explícitamente todas las semanas del periodo analizado para cada combinación producto–tienda. Las semanas sin observaciones se imputan de forma coherente con la hipótesis adoptada: QUANTITY = 0, SALES\_NET y SALES\_NO\_DISCOUNT = 0, DISC\_PCT = 0, mientras que las variables de precio (PRICE y PRICE\_BASE) se dejan como valores no definidos (NaN).

Con el panel regular, los retardos recuperan su interpretación estrictamente temporal: LAG\_1 representa la semana inmediatamente anterior (aunque la venta haya sido cero), LAG\_4 captura un ciclo mensual aproximado y LAG\_12 una referencia trimestral. Esta estructura es especialmente relevante en contextos de demanda intermitente, donde la presencia de ceros constituye una señal informativa.

Como limitación metodológica, esta decisión implica mezclar situaciones de demanda real nula con casos de demanda no observada por problemas de disponibilidad. Dado que el dataset no permite distinguir ambos escenarios, esta limitación se acepta explícitamente y se mitiga mediante filtros de modelabilidad

(presencia mínima del 40% de semanas y ventas promedio  $\geq 2$  unidades por semana), reteniendo únicamente series con señal suficiente. Aunque existen alternativas más complejas, como tratar las semanas sin registro como valores faltantes o introducir una variable de disponibilidad, en este TFM se prioriza la simplicidad, la consistencia temporal y la robustez del simulador.

# 3. Feature Engineering

El proceso de feature engineering tuvo como objetivo enriquecer la información disponible para capturar dinámica temporal, estacionalidad y heterogeneidad estructural entre productos y tiendas. Las variables creadas se agrupan en cuatro bloques: dinámica temporal (LAGs), estacionalidad, variables de contexto y variable objetivo (LIFT).

## 3.1 Variables de dinámica temporal (LAGs)

Para capturar persistencia e inercia de la demanda, se construyeron variables de rezago sobre la cantidad vendida:

- LAG\_1: Ventas de la semana anterior (persistencia inmediata).
- LAG\_2: Ventas de dos semanas atrás (suavizado de volatilidad).
- LAG\_4: Ventas de cuatro semanas atrás (aproximación a ciclo mensual).
- LAG\_12: Ventas de doce semanas atrás (aproximación a ciclo trimestral).

Dado que LAG\_12 requiere al menos doce semanas previas, las primeras observaciones de cada combinación producto–tienda fueron eliminadas para evitar valores indefinidos.

El uso de lags resulta especialmente relevante en contextos de demanda intermitente. En productos de baja rotación, la persistencia de ceros en semanas consecutivas es informativa; en productos más estables, los lags permiten capturar nivel local y dinámica de corto plazo.

## 3.2 Variables de estacionalidad

Para modelar patrones cílicos anuales, se empleó una codificación trigonométrica de la semana del año:

$$WEEK\_SIN = \sin\left(\frac{2\pi \cdot WEEK\_NO}{52}\right)$$

$$WEEK\_COS = \cos\left(\frac{2\pi \cdot WEEK\_NO}{52}\right)$$

Esta representación es ampliamente utilizada para variables temporales cílicas, ya que preserva la continuidad entre la semana 52 y la semana 1. Frente al uso de variables dummy, este enfoque:

- Reduce dimensionalidad.
- Evita colinealidad.
- Modela correctamente la naturaleza cílica del tiempo.

Adicionalmente, se incorporó la variable WEEK\_OF\_MONTH, que aproxima un ciclo de repetición cada cuatro semanas, capturando posibles patrones de reposición, efectos de calendario o cadencias promocionales.

Si bien los meses no contienen exactamente cuatro semanas, esta variable introduce señal adicional con bajo coste computacional y mejora la capacidad explicativa del modelo.

### 3.3 Variables de contexto (priors de nivel)

Los productos y tiendas presentan niveles estructurales de demanda distintos. Para capturar esta heterogeneidad se construyeron dos variables:

- **PROD\_MEAN\_SALES**: media histórica de ventas por producto.
- **STORE\_MEAN\_SALES**: media histórica de ventas por tienda.

Estas variables actúan como “anclas de nivel”, permitiendo que el Modelo B diferencie entre productos o tiendas de alta y baja rotación.

Para evitar filtración de información (data leakage):

1. Las medias se calcularon exclusivamente sobre el conjunto de entrenamiento.
2. Posteriormente se aplicaron al conjunto de test sin recalcular.
3. En caso de combinaciones no observadas, se imputó la media global del entrenamiento.

Estas variables mejoran la capacidad de generalización del modelo y estabilizan la estimación del lift en contextos con demanda heterogénea.

### 3.4 Variable objetivo: Lift incremental

El Modelo B no predice ventas totales, sino el incremento atribuible a la promoción.

$$\text{LIFT} = \text{QUANTITY} - \text{BASELINE\_PRED}$$

donde:

- **QUANTITY**: ventas observadas.
- **BASELINE\_PRED**: ventas esperadas sin promoción (estimadas por el Modelo A).

Esta formulación permite separar explícitamente:

1. La demanda estructural (capturada por el baseline).
2. El efecto incremental de la promoción.

El uso del lift como variable objetivo evita que el segundo modelo reaprenda el nivel base, concentrando su capacidad en estimar la sensibilidad promocional.

Desde una perspectiva de negocio, el lift es directamente interpretable como unidades adicionales generadas por la promoción, facilitando la toma de decisiones y la comparación entre escenarios.

### 3.5 Resumen del dataset final

El resultado de esta sección es el dataframe wk: un panel ordenado por PRODUCT\_ID, STORE\_ID y WEEK\_NO, con variables de ventas, precios, descuento, lags y estacionalidad. Este panel se utiliza como base para el entrenamiento de los modelos A y B y para la simulación de escenarios.

# 4. Modelo A (Baseline): especificación, supuestos e interpretación

El Modelo A tiene como función principal generar: ¿cuántas unidades se habrían vendido en ausencia de descuento? Para ello se ajusta una regresión OLS sobre semanas sin promoción. Esta sección entra en detalle sobre la elección de OLS, sus supuestos y su papel en el pipeline.

Se decide emplear un modelo OLS como baseline por los siguientes motivos:

- Interpretabilidad: coeficientes permiten entender el efecto marginal aproximado del precio y de los lags.
- Simplicidad: reduce riesgo de sobreajuste en un conjunto de datos con señal limitada.
- Se busca robustez más que precisión extrema.

En un escenario con datos más ricos, podrían usarse modelos más complejos (p. ej., modelos con efectos fijos), pero la elección de OLS es adecuada como primer nivel y facilita la explicación académica.

## 4.1. Submuestra de entrenamiento: semanas sin descuento y corte temporal

El entrenamiento del baseline se restringe a dos condiciones: (i)  $\text{DISC\_PCT} = 0$  (sin promoción) y (ii)  $\text{WEEK\_NO} \leq 80$  (train). Esta segunda condición evita mirar al futuro y permite evaluar generalización temporal en el test posterior.

## 4.2. Variables explicativas y su rol

El baseline incluye PRICE, lags, estacionalidad y WEEK\_OF\_MONTH. La lógica es:

- PRICE capta sensibilidad al precio en ausencia de promoción.
- Lags controlan por persistencia de demanda.
- Estacionalidad explica patrones recurrentes.
- WEEK\_OF\_MONTH añade una señal discreta adicional.

## 4.3. Tratamiento de NaN e infinitos

Al construir  $\text{PRICE} = \text{SALES\_NET}/\text{QUANTITY}$ , las semanas con  $\text{QUANTITY}=0$  generan divisiones por cero. El pipeline reemplaza infinitos por NaN y elimina filas con NaN antes de ajustar el modelo.

#### 4.4. Ajuste del modelo OLS y lectura del summary

El modelo se ajusta con statsmodels.OLS añadiendo constante. El summary de OLS proporciona coeficientes, errores estándar, t-stats y R<sup>2</sup>. En el contexto de este TFM, el interés principal es la dirección y consistencia del efecto de PRICE y la capacidad del baseline de generar predicciones razonables.

#### 4.5. Función predecir\_baseline\_ols y recorte a no negativo

Una vez entrenado, se define una función para predecir baseline sobre cualquier dataframe con las mismas features. Las predicciones se recortan a valores  $\geq 0$ , ya que las unidades vendidas no pueden ser negativas.

#### 4.6 Ajuste del Modelo A y validación del baseline

El Modelo A se estima mediante regresión lineal ordinaria (OLS) utilizando la librería ‘statsmodels’, incorporando una constante en la especificación. El resumen estadístico del modelo proporciona los coeficientes estimados, errores estándar, estadísticos t y el coeficiente de determinación (R<sup>2</sup>).

En el contexto de este trabajo, el interés no se centra únicamente en la significatividad estadística, sino en:

- La dirección del efecto del precio (PRICE), que debe ser coherente con la teoría económica.
- La consistencia de los coeficientes asociados a los rezagos.
- La capacidad del modelo para generar un baseline estable y razonable en semanas sin promoción.

Más que maximizar R<sup>2</sup>, el objetivo es obtener un contrafactual robusto que sirva como referencia para el cálculo del lift.

#### 4.7. Riesgos y limitaciones del baseline

El baseline OLS, aunque interpretable, tiene limitaciones:

- Linealidad: asume relación lineal entre features y ventas; puede ser una aproximación.
- Heterocedasticidad: el error puede variar con el nivel de ventas.
- Variables omitidas: falta información de stock, competencia u otras palancas.
- Intermitencia: series con muchos ceros reducen capacidad de ajuste.

Estas limitaciones se mitigan parcialmente al usar el baseline como intermediario (en lugar de predicción final) y al delegar en el Modelo B la captura de no linealidades asociadas a promociones.

# 5. Modelo B (Lift): aprendizaje no lineal del efecto promocional

El Modelo B se plantea como un modelo incremental cuyo objetivo es predecir el lift en unidades, definido como la diferencia entre las ventas observadas y el baseline estimado por el Modelo A:

$$LIFT = QUANTITY - BASELINE\_PRED$$

De esta forma, el Modelo B se centra específicamente en capturar el efecto adicional atribuible a la promoción, separándolo del nivel base de demanda. Para ello, se construye un conjunto de variables explicativas (features\_B) que combina señales promocionales, contexto de demanda, dinámica temporal y heterogeneidad estructural entre productos y tiendas.

En concreto, el set de features incluye: BASELINE\_PRED, DISC\_PCT, lags (LAG\_1, LAG\_2, LAG\_4, LAG\_12), estacionalidad (WEEK\_SIN, WEEK\_COS), WEEK\_OF\_MONTH y “priors” de producto y tienda (PROD\_MEAN\_SALES, STORE\_MEAN\_SALES)

## 5.1 Variables promocionales y de intensidad: DISC\_PCT

La variable DISC\_PCT representa el porcentaje de descuento aplicado (acotado en [0, 0.9]) y actúa como el principal “driver” del efecto promocional. Esta feature permite al modelo aprender cómo cambia el lift conforme aumenta o disminuye el descuento, capturando posibles no linealidades (por ejemplo, rendimientos decrecientes: pasar de 10% a 20% puede tener más impacto que de 30% a 40%, dependiendo del producto). Al incluir DISC\_PCT directamente en el modelo incremental, se fuerza a que el aprendizaje se focalice en la relación *descuento → incremento de unidades* en lugar de mezclarlo con el nivel base.

## 5.2 Interacción implícita con el nivel base: BASELINE\_PRED

BASELINE\_PRED incorpora el nivel de demanda esperada sin promoción estimado por el Modelo A. Su presencia en el Modelo B es clave porque el efecto de una promoción suele depender del punto de partida: no es igual aplicar un 20% de descuento a un producto con ventas base altas (donde puede haber más volumen incremental) que a uno con ventas base muy bajas (donde el lift puede ser limitado o más errático). En términos prácticos, BASELINE\_PRED funciona como una variable de contexto que permite al modelo modular el lift según la “capacidad” de demanda natural del producto en esa tienda y semana.

### 5.3 Dinámica reciente y memoria de la serie: lags LAG\_1, LAG\_2, LAG\_4, LAG\_12

Las variables de retardo (LAG\_1, LAG\_2, LAG\_4, LAG\_12) capturan el estado reciente de la serie de ventas y aportan información sobre inercia, tendencias de corto plazo y patrones repetitivos. Por ejemplo:

- LAG\_1 / LAG\_2: recogen la demanda inmediata y ayudan a incorporar shocks recientes (roturas de stock, picos locales, efectos residuales de campañas anteriores).
- LAG\_4: aproxima comportamiento “mensual” o de ciclo corto, útil cuando hay patrones de compra repetitivos cada poca semana.
- LAG\_12: aporta memoria más larga y puede aproximar ciclos trimestrales o patrones de estacionalidad más amplios.

En un modelo de lift, estos lags no solo ayudan a predecir el nivel esperado, sino que permiten aprender cuándo una promo es más efectiva: por ejemplo, si un producto venía en crecimiento, el lift podría ser mayor; si venía cayendo, el lift podría ser más bajo.

### 5.4 Calendario y estacionalidad: WEEK\_SIN, WEEK\_COS y WEEK\_OF\_MONTH

La demanda retail suele presentar componentes estacionales (semanales, mensuales, etc.). Para capturarla sin introducir discontinuidades, se usa una representación cíclica mediante WEEK\_SIN y WEEK\_COS, calculadas a partir de la semana del año. Este enfoque permite que el modelo identifique patrones periódicos (por ejemplo, semanas próximas a festivos o momentos del año con mayor consumo), evitando el problema de tratar “semana 52” y “semana 1” como lejanas. Además, WEEK\_OF\_MONTH añade estructura intra-mes, útil para detectar patrones tipo “principio de mes vs final de mes” (por ejemplo, efectos de nómina o ciclos de reposición). Con ello, el Modelo B puede aprender que la efectividad de un descuento puede variar según el calendario, incluso para el mismo producto y tienda.

### 5.5 Priors de producto y tienda: PROD\_MEAN\_SALES y STORE\_MEAN\_SALES

Para capturar heterogeneidad estructural, se incluyen dos variables de contexto construidas como medias históricas en train:

- PROD\_MEAN\_SALES: media de ventas del producto (en el histórico de entrenamiento).
- STORE\_MEAN\_SALES: media de ventas de la tienda (en el histórico de entrenamiento).

Estas variables actúan como priors o “anclas” estadístico-operativas que ayudan al modelo a estabilizar predicciones, especialmente en entornos con demanda ruidosa o intermitente. En la práctica, permiten que el Modelo B distinga entre (i) productos intrínsecamente de alta rotación frente a baja rotación y (ii) tiendas con mayor o menor volumen general. Además, en el conjunto de test se contempla el caso de productos/tiendas no vistas rellenando con una media global, evitando NaNs y manteniendo coherencia en inferencia.

## 5.6 Resultado: un set de features “explicativo” del lift

En conjunto, estas features permiten al Modelo B aprender simultáneamente:

- La respuesta del lift al descuento (DISC\_PCT) y su posible no linealidad.
- Cómo depende esa respuesta del nivel base (BASELINE\_PRED), capturando que el mismo descuento puede tener efectos distintos según el contexto de demanda.
- Cómo modula el efecto promocional la dinámica reciente (lags) y el calendario (WEEK\_SIN/COS, WEEK\_OF\_MONTH).
- Cómo varía la efectividad por producto y por tienda mediante priors agregados (PROD\_MEAN\_SALES, STORE\_MEAN\_SALES) que incorporan heterogeneidad estructural.

Finalmente, al modelar directamente el incremento (LIFT) en vez de la venta total, se reduce el riesgo de que el modelo confunda cambios debidos a estacionalidad o nivel base con el impacto real de la promoción, logrando un enfoque más interpretable y operativo para simulación de escenarios.

## 5.7. Modelos candidatos y justificación

Se comparan tres enfoques:

- **Regresión lineal**, utilizada como benchmark.
- **XGBoost**, basado en boosting de árboles de decisión.
- **CatBoost**, variante optimizada para variables categóricas y relaciones no lineales complejas.

La regresión lineal sirve como referencia simple e interpretable, mientras que los modelos de boosting aportan:

- Capacidad para modelar no linealidades.
- Captura automática de interacciones.
- Mayor flexibilidad en presencia de relaciones complejas entre descuento y ventas.

## 5.8. Hiperparámetros y trade-offs

Los hiperparámetros controlan el trade-off sesgo-varianza: más árboles y mayor profundidad aumentan capacidad, pero también riesgo de sobreajuste. En el código se eligen valores moderados (400 iteraciones), adecuados como baseline.

## 5.9. Evaluación: MAE en ventas totales

Aunque el modelo predice LIFT, la evaluación se realiza en ventas totales:  
**ventas\_pred = BASELINE\_PRED + lift\_pred.**

Se utiliza el MAE como métrica principal por tres razones:

1. Se expresa directamente en unidades vendidas.
2. Es fácilmente interpretable desde una perspectiva de negocio.
3. Es robusto ante valores atípicos moderados.

Una figura comparativa de MAE entre modelos (Linear, XGBoost y CatBoost) en el conjunto de test permite identificar el modelo con mejor desempeño.

### 5.10 Consideraciones sobre sesgo del baseline

Dado que el lift se calcula restando BASELINE\_PRED, cualquier sesgo del baseline se traslada al lift. Por ejemplo, si el baseline sobreestima sistemáticamente, el lift será más negativo. Por ello, es importante validar el baseline en semanas sin promoción (por ejemplo, MAE baseline) y revisar su estabilidad antes de confiar en el lift.

### 5.11 Robustez y generalización

En problemas temporales, la robustez se refuerza con validación por ventanas deslizantes (rolling-origin). El código usa un único split por simplicidad; como mejora futura se sugiere replicar el entrenamiento en varias ventanas y promediar métricas.

# 6. Simulador de escenarios promocionales: diseño, uso y limitaciones

Una de las principales aportaciones del trabajo es transformar el modelo predictivo en una herramienta de soporte a la decisión comercial. En lugar de quedarse en una métrica de precisión o en una predicción puntual, el simulador permite responder a la pregunta más relevante para negocio: qué pasaría con las ventas y el margen si cambiamos el nivel de descuento. En términos prácticos, convierte el modelo en un “probador de escenarios” que ayuda a planificar promociones, comparar alternativas y priorizar productos con mayor potencial, de forma rápida y consistente.

El simulador está pensado para que un usuario de negocio pueda explorar escenarios sin necesidad de reentrenar modelos ni construir manualmente variables. A partir de un producto, una tienda (o el agregado de todas) y una semana objetivo, el usuario indica un porcentaje de descuento y el sistema devuelve una estimación de unidades esperadas. Ese resultado no es solo una cifra final: internamente se descompone en dos piezas con interpretación clara. Por un lado, estima la demanda “normal” o base que el producto tendría sin promoción. Por otro, estima el incremento incremental atribuible a la promoción (uplift). Esta separación es especialmente útil en negocio porque permite distinguir entre productos que venden bien por sí mismos y productos que dependen de la promoción para generar volumen adicional.

La lógica interna se apoya en el histórico de ventas del propio producto en esa tienda (o en las tiendas elegibles cuando se agrega). El primer paso consiste en recuperar el historial disponible hasta la semana anterior a la que se quiere simular. A partir de ese histórico se calculan señales que representan la inercia de la demanda: ventas de semanas recientes (lags) que capturan tanto la tendencia inmediata como patrones más estables. Para que el simulador sea mínimamente fiable, se requiere un histórico suficiente, ya que algunas de estas señales necesitan mirar varias semanas hacia atrás. En la práctica, el requisito clave es disponer de al menos 12 observaciones previas para poder construir el lag de 12 semanas sin inventar valores.

Además de la memoria de la demanda, se incorporan variables de calendario que recogen la estacionalidad. En retail, el “momento” del año importa: hay semanas más fuertes y semanas más débiles por hábitos de compra, eventos y patrones cíclicos. Por eso, el simulador utiliza una codificación estacional basada en la semana del año y la posición dentro del mes, lo cual permite que una misma promoción tenga impactos distintos según el periodo.

Un punto importante, desde el punto de vista de negocio, es el tratamiento del precio. El simulador no intenta adivinar políticas futuras de precio base; necesita un valor representativo para estimar la demanda sin promoción. Por simplicidad y robustez, se utiliza un precio representativo del histórico (por ejemplo, la mediana). Esto permite simular el efecto del descuento sobre un contexto de precio razonable, pero también implica una condición: si en el futuro cambia la estrategia de precio base (o hay cambios estructurales de PVP), el resultado puede desviarse.

Con todas esas variables, el simulador ejecuta una predicción en dos pasos. Primero calcula la venta esperada sin promoción (baseline), que representa la “línea de referencia” sobre la que se evalúa la promo. Después estima el incremento por promoción (lift) en función del descuento y del contexto (nivel base, dinámica reciente, calendario y priors de producto/tienda). Finalmente, suma baseline y lift para obtener la predicción de unidades, aplicando un recorte a cero para evitar resultados negativos que no tienen sentido en unidades.

## 6.1 Cómo interpretar el resultado

El resultado del simulador debe interpretarse como una expectativa bajo condiciones similares a las observadas históricamente. El valor final responde a la pregunta: con este descuento, en esta semana y en este ámbito (tienda o agregado), cuántas unidades se esperan vender. La descomposición baseline + lift es especialmente actionable:

- Baseline: indica el volumen que se esperaría sin promoción. Sirve para entender la “salud” del producto y su rotación natural.
- Lift: indica el volumen incremental atribuible a la promoción. Sirve para valorar si la promoción merece la pena y para comparar descuentos alternativos.

En la práctica, la utilidad del simulador no es solo elegir “si promocionar o no”, sino comparar escenarios: por ejemplo, un 10% frente a un 20% o un 30%. Es habitual que el incremento en unidades no crezca linealmente con el descuento: puede haber tramos con rendimientos decrecientes (subir del 20% al 30% aporta poco volumen adicional) o comportamientos más sensibles en determinados productos. El simulador permite ver ese trade-off y discutirlo con negocio en términos de impacto.

Para facilitar su adopción, el simulador se ha implementado con una interfaz sencilla: selección de producto, selección de tienda (incluyendo la opción de todas), un control de descuento y un botón de ejecución. El objetivo es permitir una exploración rápida, tipo “what-if”, sin necesidad de escribir código ni manipular datasets. Este formato encaja bien en sesiones de planificación comercial, donde se discuten distintos niveles de descuento y se necesita una referencia cuantitativa inmediata.

En escenarios reales, una recomendación práctica es acompañar el simulador con una visualización de histórico reciente y las predicciones de las próximas semanas, de forma que el usuario vea el contexto: si el producto tiene tendencia creciente, si está muy intermitente, o si el baseline es prácticamente cero. Esto ayuda a interpretar resultados y evita tomar decisiones sobre señales poco fiables.

## 6.2 Agregación “TODAS”: cómo se obtiene y por qué se restringe

Cuando el usuario selecciona TODAS, el simulador no “promedia” sin más. La lógica es más operativa: calcula una predicción por tienda y después agrega. Este enfoque tiene dos ventajas para negocio. La primera es que respeta la heterogeneidad real: una promoción puede funcionar mejor en unas tiendas que en otras por perfil de cliente o nivel de demanda. La segunda es que produce un agregado que tiene sentido como suma de volúmenes.

Sin embargo, para evitar resultados artificiales, no se incluyen todas las tiendas indiscriminadamente. Solo se consideran tiendas con histórico suficiente para el producto, ya que, si falta histórico, las variables de memoria (lags) no se pueden construir con fiabilidad y la predicción se vuelve inestable. Por tanto, el agregado TODAS debe interpretarse como “todas las tiendas elegibles con histórico suficiente”, no necesariamente todas las tiendas existentes en el dataset. Esto es importante a nivel de negocio porque el volumen agregado dependerá del número de tiendas elegibles: si el producto está poco distribuido o tiene poca historia en muchas tiendas, el agregado representará un subconjunto.

## 6.3 Cálculo del margen en el simulador

Además de estimar unidades vendidas, el simulador incorpora el cálculo de ingresos, coste y margen para cada escenario de descuento. Esta ampliación es especialmente relevante desde el punto de vista de negocio: una promoción no debe evaluarse únicamente por su capacidad de aumentar el volumen, sino por su capacidad de generar resultado económico. En retail es frecuente que un descuento incremente las unidades, pero reduzca el margen unitario; por tanto, la pregunta clave no es “qué descuento vende más”, sino “qué descuento maximiza el margen (o el margen porcentual) manteniendo un volumen razonable”.

Desde el punto de vista técnico, el simulador genera primero una predicción de unidades para cada escenario. Esa predicción se obtiene con el enfoque baseline + lift: el Modelo A estima la demanda sin promoción (baseline) y el Modelo B estima el incremento atribuible al descuento (lift). La suma de ambos produce la predicción final de unidades, recortada a cero para evitar valores negativos.

Una vez estimadas las unidades, el simulador traduce esa predicción a métricas económicas. Para ello utiliza una estimación del precio base sin descuento

(PRICE\_BASE) y calcula el precio efectivo aplicando el descuento seleccionado. En el código, el precio efectivo se define como:

$$\text{precio\_efectivo} = \text{price\_base} \times (1 - \text{descuento})$$

A partir de ahí se calculan los ingresos previstos del escenario como:

$$\text{ingresos} = \text{unidades\_predichas} \times \text{precio\_efectivo}$$

Este paso es importante porque separa el volumen (unidades) del efecto directo del descuento sobre el ingreso unitario. En otras palabras: aunque suban las unidades, el ingreso por unidad baja cuando aumenta el descuento, y ambos efectos compiten entre sí.

El segundo componente necesario para calcular margen es el coste. En el simulador se asume un coste proporcional al precio base, mediante el parámetro COST\_PCT. En la implementación se fija, por defecto, COST\_PCT = 0.70, interpretado como “el coste equivale al 70% del precio base sin descuento”. Así, el coste total del escenario se calcula como:

$$\text{coste} = \text{unidades\_predichas} \times (\text{COST\_PCT} \times \text{price\_base})$$

Con ingresos y coste, el margen absoluto se obtiene como:

$$\text{margen} = \text{ingresos} - \text{coste}$$

Y para completar el análisis, también se calcula el margen porcentual (margen sobre ingresos):

$$\text{margen\_pct} = 100 \times \text{margen} / \text{ingresos} \text{ (si } \text{ingresos} > 0\text{)}$$

Esta lógica permite reportar tanto el margen en euros como la eficiencia del descuento en términos de porcentaje, que en negocio puede ser igual o más importante dependiendo del objetivo (maximizar beneficio total, proteger rentabilidad, o mantener un margen mínimo). En tu simulador estas fórmulas están implementadas explícitamente en la función de cálculo financiero y se aplican para cada escenario de descuento.

Un aspecto diferencial del diseño es que el cálculo de margen se realiza de forma coherente con la predicción final de unidades por escenario, y no como un post-proceso superficial. Esto es especialmente importante cuando se trabaja con el agregado TODAS. En el modo agregado, el simulador no calcula una predicción “global” directa, sino que predice por tienda y semana, y después suma. Este enfoque tiene dos ventajas: respeta la heterogeneidad real de tiendas (no todas responden igual a una promo) y evita errores cuando algunas tiendas no tienen histórico suficiente. En tu implementación, además, se introduce un mecanismo para reforzar la coherencia económica: se corrige la monotonía de unidades por tienda y semana ( $\text{baseline} \leq d - \Delta \leq d \leq d + \Delta$ ) mediante un ajuste tipo cumulative max, y después se recalculan ingresos, costes y márgenes con las unidades ya ajustadas. Esto garantiza

que el cálculo de margen esté alineado con el comportamiento final que se comunica al usuario.

Desde el punto de vista del negocio, incorporar margen al simulador cambia por completo el uso de la herramienta. Si solo se muestran unidades, la recomendación suele sesgar hacia descuentos altos, porque en muchos productos el modelo tenderá a predecir más volumen al aumentar el descuento. Sin embargo, a medida que el descuento sube, el precio efectivo cae y el margen unitario se estrecha. Llega un punto en el que vender más no compensa la pérdida de margen por unidad; incluso puede ocurrir que el margen total disminuya, aunque el volumen suba. El simulador permite identificar ese punto de equilibrio con claridad.

Esto se puede interpretar como un trade-off natural entre dos objetivos:

- Objetivo de volumen: maximizar unidades para ganar cuota, acelerar rotación o liquidar inventario.
- Objetivo de rentabilidad: maximizar margen total o mantener un margen porcentual mínimo.

Con la capa de margen, el simulador deja de ser una herramienta de predicción y pasa a ser una herramienta de decisión: permite comparar escenarios y elegir el descuento óptimo según el criterio de negocio. En tu interfaz se materializa mediante la opción de “Optimizar por margen total” o “Optimizar por margen %”, y el sistema marca automáticamente el mejor escenario dentro del horizonte simulado.

Para ilustrarlo conceptualmente, la lógica de decisión puede expresarse así: en un horizonte H de semanas, para cada escenario de descuento se calcula la suma de margen (y unidades) y se selecciona el escenario con mayor valor según el objetivo. Este enfoque es muy útil cuando el negocio tiene que decidir entre varias mecánicas promocionales parecidas (por ejemplo, 10%, 20% o 30%) y quiere justificar por qué una opción es “mejor” más allá del volumen.

La aportación práctica más importante es que el simulador permite responder preguntas típicas de planificación comercial con métricas alineadas con el P&L:

- Si subo el descuento de 10% a 20%, ¿cuánto margen adicional espero ganar (o perder)?
- ¿Cuál es el descuento que maximiza margen total en las próximas semanas?
- ¿Qué escenario mantiene mejor el margen porcentual si mi objetivo es proteger rentabilidad?
- ¿Cuánto volumen extra compro con cada punto de descuento y a qué coste en margen?

También facilita conversaciones entre áreas: comercial puede priorizar unidades, finanzas puede priorizar margen, y supply puede considerar rotación e inventario. Al

tener unidades e indicadores económicos en la misma salida, la discusión se hace más transparente y basada en cuantificación, no en intuición.

Por último, conviene explicitar las hipótesis y limitaciones del cálculo de margen, ya que condicionan la interpretación. La principal simplificación es que el coste se aproxima como un porcentaje fijo del precio base (COST\_PCT). Esto es razonable como aproximación cuando no se dispone de coste real por SKU, pero puede no capturar variaciones reales (costes logísticos, cambios de proveedor, diferencias por formato, etc.). Asimismo, el precio base se estima a partir del histórico (por ejemplo, mediana), lo que implica que el simulador asume estabilidad de precios base; si el PVP base cambia por inflación o reposicionamiento, el margen estimado puede desviarse. Aun así, incluso con estas simplificaciones, la capa de margen aporta un valor claro: convierte el análisis promocional en una comparación económica consistente entre alternativas, evitando recomendar descuentos altos solo porque “venden más”.

En resumen, incorporar el cálculo de margen permite que el simulador no se limite a estimar ventas, sino que apoye decisiones de promoción con criterios de rentabilidad. En un entorno real, esto es crucial: el éxito de una promoción no se mide por unidades, sino por el equilibrio entre volumen incremental y contribución al margen, dentro de los objetivos comerciales y las restricciones operativas.

#### 6.4 Limitaciones y precauciones de uso

Como cualquier herramienta predictiva, el simulador tiene limitaciones que conviene explicitar para evitar un uso incorrecto:

1. Extrapolación de descuentos. Si se simulan descuentos muy altos y poco observados en el histórico, el resultado puede ser menos fiable. El modelo aprende de lo que ha visto; fuera de ese rango, la relación descuento-demanda puede no sostenerse.
2. Estabilidad del contexto. El simulador asume que el futuro se parece al pasado en términos de comportamiento de compra y condiciones del entorno. Cambios estructurales (cambio de surtido, reposicionamiento de marca, cambios de distribución, campañas externas, competidores) pueden alterar la respuesta real.
3. Precio base representativo. El baseline se estima con un precio “típico” del histórico. Si el precio base cambia en el futuro (por inflación, cambio de estrategia o reposicionamiento), la referencia sobre la que se calcula el lift puede quedar desalineada.

4. Variables no observadas. El simulador no incorpora restricciones de stock, roturas, disponibilidad en lineal, acciones de competencia o eventos externos. En negocio, estas variables pueden dominar el resultado real.
5. Demanda intermitente. En productos de baja rotación o muy intermitentes, pequeñas variaciones generan cambios relativos grandes. En esos casos, el simulador puede mostrar respuestas no intuitivas o inestables, por lo que conviene usarlo como guía y no como verdad exacta.

En resumen, el simulador actúa como un puente entre analítica y negocio: permite traducir un modelo de datos en un flujo de decisión práctico, centrado en el impacto esperado de promociones. Su uso recomendado es como herramienta de planificación y priorización, complementada con criterios comerciales y limitaciones operativas, especialmente en categorías con demanda irregular o cuando se simulan descuentos fuera del patrón histórico.

# 7. Validación, diagnóstico y sistema de recomendación

Además de entrenar modelos, es crucial validar que sus predicciones son razonables. Esta sección documenta las utilidades implementadas en el código: backtesting con métricas, diagnóstico de monotonía y ranking de productos basado en confianza.

## 7.1. Backtesting (CHECK): Real vs Pred

El bloque `run_check` evalúa el simulador sobre semanas pasadas. Para un producto y una tienda (o TODAS) se generan predicciones usando el descuento real observado esa semana, y se comparan con ventas reales.

Se excluyen las primeras 12 semanas evaluables para garantizar que existen lags suficientes. Esto evita resultados erróneos por falta de histórico.

## 7.2. Diagnóstico de monotonía

La monotonía es una propiedad de sentido común: si se aumenta el descuento manteniendo el resto constante, las ventas esperadas deberían aumentar o, al menos, no disminuir. El código evalúa varios escenarios (por ejemplo, 10%, 20%, 30%) y marca `MONOTONIC_OK` si se cumple el orden.

Este check no impone monotonía en el entrenamiento; funciona como auditoría para detectar productos donde el modelo es inestable o donde el histórico no soporta una relación clara.

## 7.3. Productos problemáticos y causas típicas

El diagnóstico etiqueta como problemáticos aquellos con:

- No monotonicidad (predicciones inconsistentes al variar el descuento).
- Lift negativo (promoción reduce ventas), que puede reflejar ruido o canibalización.
- Baseline casi nulo (poca señal), donde el modelo es más volátil.

En la práctica, estos casos suelen coincidir con productos de demanda intermitente, con pocas observaciones o con promociones raras (muy pocas semanas con descuento).

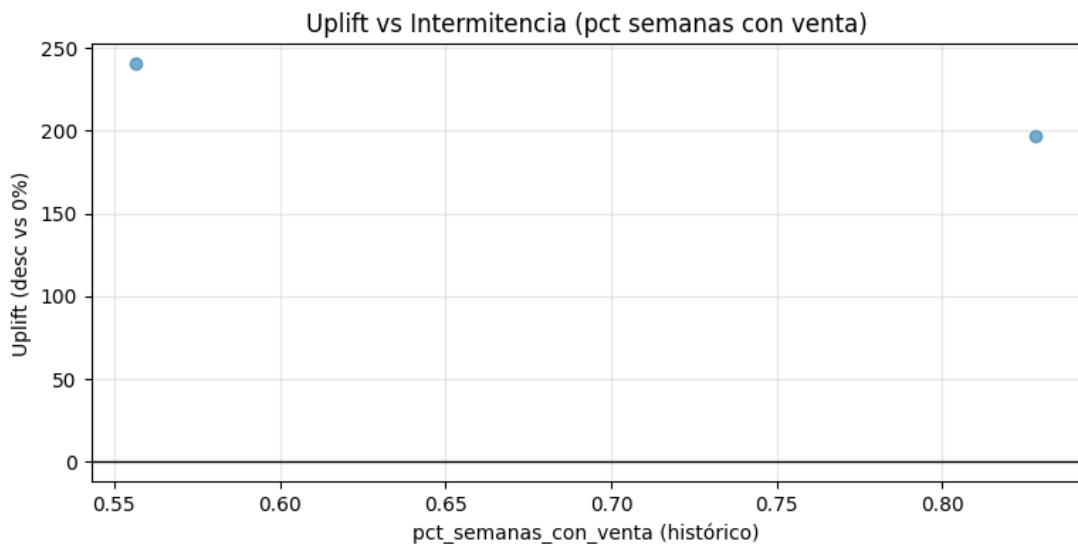
## 7.4. Score de confianza (0–1): componentes

Para priorizar productos a promocionar, no basta con ordenar por uplift: conviene ponderar por la fiabilidad de la estimación. Se define un score de confianza multiplicativo compuesto por cuatro factores:

- `conf_monotonic`: 1 si monótonico, 0 si no.

- **conf\_base**: escala con PRED\_0 (ventas sin descuento).
- **conf\_sale\_freq**: escala con pct de semanas con venta.
- **conf\_weeks**: escala con weeks\_obs (número de semanas observadas).

Se usa un enfoque multiplicativo para que un fallo grave (p. ej., no monotonicidad) reduzca la confianza total de forma contundente.



## 7.5. Ranking final y decisión

El ranking se define como:  $\text{RANK\_SCORE} = \text{UPLIFT\_vs\_0} \times \text{CONF\_SCORE}$ . Esto produce una lista de productos recomendados donde el impacto esperado se ajusta por fiabilidad.

TOP recomendados por RANK_SCORE (uplift * confianza):											
	PRODUCT_ID	STORE	PRED_0	UPLIFT_vs_0	CONF_SCORE	RANK_SCORE	MONOTONIC_OK	LIFT_NEGATIVE	BASE_TOO_LOW	pct_weeks_sale	weeks_obs
0	1082185	TODAS(114)	347.49	196.67	0.754703	148.427409	True	False	False	0.828292	87
1	995242	TODAS(114)	129.91	240.62	0.366664	88.226640	True	False	False	0.556665	87

## 7.6. Cómo usar el ranking en un caso real

1. Filtrar productos con CONF\_SCORE por encima de un umbral (p.ej., 0.3).
2. Ordenar por RANK\_SCORE y seleccionar top-k para análisis.
3. Para cada candidato, ejecutar el simulador con varios descuentos y revisar la curva de respuesta.
4. Aplicar restricciones de negocio (margen, stock, estrategia de categoría).
5. Monitorizar resultados reales y retroalimentar el modelo.

# 8. Resultados, conclusiones, limitaciones y trabajo futuro

## 8.1. Resultados: qué se obtiene al ejecutar el pipeline

Al ejecutar el notebook/script completo se obtiene: Un OLS baseline con summary estadístico, métricas MAE de predicción total para distintos modelos de lift, un simulador funcional, herramientas de check y un ranking priorizado.

En el código se imprime el MAE comparativo de los modelos (Linear, XGBoost, CatBoost) sobre la predicción total (baseline+lift). Para documentar resultados en la memoria final, se recomienda capturar esas salidas y volcarlas como tablas o figuras.

## 8.2. Principales conclusiones

- Separar baseline y lift mejora la interpretabilidad del modelo promocional.
- El baseline entrenado solo en semanas sin descuento actúa como contrafactual útil.
- Los modelos de boosting capturan no linealidades e interacciones relevantes para el lift.
- El simulador y el ranking convierten el modelo en una herramienta de soporte a decisiones.

## 8.3. Limitaciones

- Intermitencia de demanda: reduce número de productos modelables y aumenta varianza.
- Variables omitidas: sin stock, competencia o exposición, el modelo puede atribuir a descuento efectos de otros factores.
- Supuesto de precio representativo: usar la mediana histórica puede no reflejar cambios futuros.
- Validación simplificada: un único split temporal; se recomienda validación por ventanas.

## 8.4. Trabajo futuro

- Incorporar variables de causal\_data (display/mail) y analizar sinergias.
- Modelado jerárquico o efectos fijos por producto/tienda para compartir señal.
- Imponer monotonicidad mediante restricciones en boosting.
- Extender a métricas económicas (margen/ROI) con costes imputados o reales.
- Diseñar experimentos causales (diff-in-diff) para aislar efectos con mayor rigor.

# Anexos

## Anexo A. Diccionario ampliado de variables

Este diccionario resume variables del panel wk y su interpretación:

- WEEK\_NO: número de semana (entero).
- PRODUCT\_ID: identificador de producto.
- STORE\_ID: identificador de tienda.
- QUANTITY: unidades vendidas en la semana (agregado).
- SALES\_NET: ventas netas (en este pipeline coincide con SALES\_VALUE).
- RETAIL\_DISC\_AMT: descuento retail medio semanal.
- SALES\_NO\_DISCOUNT: ventas reconstruidas sin descuento ( $SALES\_NET + |RETAIL\_DISC\_AMT|$ ).
- PRICE: precio neto medio semanal ( $SALES\_NET/QUANTITY$ ).
- PRICE\_BASE: precio base medio semanal ( $SALES\_NO\_DISCOUNT/QUANTITY$ ).
- DISC\_PCT: porcentaje de descuento (recortado a 0..0.9).
- LAG\_1/2/4/12: ventas de semanas anteriores.
- WEEK\_SIN/WEEK\_COS: componentes estacionales.
- WEEK\_OF\_MONTH: proxy mensual.
- BASELINE\_PRED: predicción del modelo OLS.
- LIFT: target del Modelo B.
- PROD\_MEAN\_SALES / STORE\_MEAN\_SALES: medias históricas (train) usadas como priors.

### Código Fuente:

[Código fuente en GitHub](#)

[Código Fuente en Google Drive](#)

### Video defensa:

[Video en GitHub](#)

[Video defensa en Google Drive](#)

### Tablas para código:

[Fichero .csv de kaggle para cargar en código](#)

### Dataset:

[Dataset Kaggle](#)

