# RESUMEN RETO 2

## 1.- CONTEXTO DE NEGOCIO Y OBJETIVO IMPLÍCITO

El dataset ***Store.csv*** contiene información de **ventas al cliente final**:

- **Pedido**: Order Date, Ship Date, Ship Mode, Order ID.
- **Cliente**: Customer Name, Segment.
- **Geografía**: City, State, Country, Region.
- **Producto**: Category, Sub-Category, Product Name.
- **Métricas económicas**: Sales, Quantity, Discount, Profit.

El enfoque es entender la **rentabilidad del negocio** por producto, cliente y región y tiempo, además dejar preparado un dataset para **modelizar el Profit** (Beneficios) con técnicas de ML.
No se estanca en "mirar ventas", sino en **vincular ventas**, **coste**, **margen**, **descuentos y logística** para explicar los resultados obtenidos.

## 2.- PREPARACIÓN DE DATOS Y PRIMEROS ACERCAMIENTOS DE NEGOCIO

**2.1- Carga del CSV** con parseo de fechas (**Order Date**, **Ship Date**).

**2.2-** Primera **revisión básica de calidad**:

- ***df.info()***, ***df.describe()*** para entender tipos y rangos.
- Comprobación de **duplicados y nulos** (***df.duplicated().sum()***, ***df.isna().sum()***).
- Conclusión: el dataset está "limpio" para poder trabajar sin imputaciones complejas.

Es decir, verificamos que la base de datos operativa de la tienda es consistente antes de apoyar decisiones sobre ella.

## 3.- FEATURE ENGINEERING ORIENTADO A NEGOCIO

Aquí introducimos la parte potente de la lógica de negocio, en donde creamos variables que traducen conceptos financieros y operativos.

**3.1- Coste y margen de beneficio**

```
df['Cost']=df['Sales']-df['Profit']
df['profit_margin']=(df['Profit']/df['Sales'])*100
```

- **Cost**: estimación del coste asociado a la venta.
- **profit_margin**: margen porcentual por línea de pedido.

Pasamos de mirar "cuánto vendo" a "cuánto gano por cada euro vendido".

**3.2- Indicador binario de ganancia / pérdida**

```
df['is_gain']=np.where(df['Profit']>0,1,0)
```

- 1 si la operación genera beneficio, 0 si genera pérdida.
  Esto permite:
- Analizar **qué productos**, **clientes o segmentos destruyen valor**.
- Hacer más adelante modelos de clasificación tipo "**pedido rentable vs pedido no rentable**".

**3.3- Tiempo de envío (calidad del servicio y logística)**

```
df['Days_to_ship']=(df['Ship Date']-df['Order Date']).dt.days.astype(int)
```

- Mide los días entre pedido y envío.
- Clave para analizar experiencia de cliente por **Ship Mode** y su posible relación con ventas y beneficio.

**3.4- Precio medio por unidad**

```
df['sales_per_quantity']=df['Sales']/df['Quantity']
```

- Aprox. al **precio unitario efectivo**.
- Sirve para ver si se está vendiendo muy barato o muy caro respecto a otras combinaciones de producto / segmento.

**3.5- Variables temporales**

```
df['month_order']=df['Order Date'].dt.month
df['year_order']=df['Order Date'].dt.year
```

Y más adelante:

```
df['Order Year'] = df['Order Date'].dt.year
df['Order Month'] = df['Order Date'].dt.month
df['Order DayOfWeek'] = df['Order Date'].dt.dayofweek
```

- Permiten analizar **tendencias por año**, **estacionalidad mensual y patrón semanal**.

Las dos primeras se pueden eliminar para evitar duplicados, ya que fueron creadas al principio como ejemplo.

**3.6- Bandas de descuento (segmentación de pricing)**

```
df['Discount_Level'] = pd.cut(
    df['Discount'],
    bins=[-0.001,0.1,0.3,1],
    labels=['Bajo','Medio','Alto']
)
```

- Convierte el descuento numérico en una **categoría de negocio**.
- Útil para responder preguntas tipo:
  "¿Los descuentos altos se compensan con volumen o simplemente hunden el margen?"

En resumen, el feature engineering se enfoca en **traducir variables transaccionales en indicadores económicos y operativos interpretables**.

## 4.- EDA ENFOCADO A LA RENTABILIDAD Y MIX DE NEGOCIO

Se realiza un EDA bastante sistemático, pero siempre con foco en negocio:

**4.1-Visión global de la cuenta de resultados**

- Suma total de **Sales**, **Profit**, **Cost**, **Quantity**, **Discount**.
- Cálculo del **profit_margin total**.
- Gráfico de barras para ver de manera rápida:
  - cuánto se factura.
  - cuánto se gasta.
  - cuánto se gana.

Esto nos dice si el negocio en conjunto es rentable o no.

**4.2- Análisis por producto y categoría**

- Agrupaciones / pivot tables:
  - **Category** y **Sub-Category** vs **Sales** y **Profit**.
  - **Product Name:** top 20 por ventas y por beneficios (y también los peores).
- Gráficos de barras de subcategorías por ventas y beneficios.

Lógica de negocio:

- Identificar **líneas de producto "estrella" y "problemáticas"**.
- Ver subcategorías que venden mucho pero aportan poco o incluso **pierden dinero**.

**4.3- Análisis de clientes**

- **top_20_customer** por suma de **Sales**.
- En un contexto real, esto sirve para:
  - Ver **clientes clave**.
  - Detectar dependencia excesiva de pocos clientes.

**4.4- Perspectiva temporal**

- **sales_for_year_cat**: ventas por año y categoría.
- **sales_for_month**: ventas por mes y año (**month_order** / **Order Month**, **year_order** / **Order Year**).
- Gráficos lineales por año / categoría.

Preguntas de negocio que se responden:

- "¿Estamos creciendo año a año?".
- "¿Hay categorías que se aceleran o se frenan?".
- "¿Hay estacionalidad clara en algunos productos?".

**4.5- Geografía: ciudad, estado, región**

- Top 15 ciudades y estados por ventas.
- Distribución de ventas por región (gráfico de pastel).
- Beneficio por región.

Lógica de negocio:

- Detectar **regiones estratégicas**, mercados maduros vs mercados emergentes.
- Identificar geográficamente dónde se genera la rentabilidad.

## 5.- DESCUENTOS, CANTIDAD Y CORRELACIONES: ENTENDER PALANCAS DE MARGEN

Entramos en relaciones clave:

**5.1- Descuento vs Profit / Sales**

```
sns.scatterplot(data=df_used,x='Discount',y='Profit',hue='Category',palette='YlGnBu')
plt.title('Discount vs Profit')
plt.show()
```

```
sns.scatterplot(data=df_used,x='Discount',y='Sales',hue='Category',palette='YlGnBu')
plt.title('Discount vs Sales')
plt.show()
```

- Visualmente permite ver si:
  - los descuentos altos están asociados a pérdidas.
  - o si realmente empujan las ventas lo suficiente.

**5.2- Quantity vs Profit**

```
sns.scatterplot(data=df_used,x='Quantity',y='Profit',hue='Category',palette='Greens')
plt.title('Quantity vs Profit')
plt.show()
```

- Comprueba si vender muchas unidades siempre implica más beneficio o si hay efectos de descuento / coste que lo contradicen.

**5.3- Matriz de correlación**

```
sns.heatmap(df_used.corr(numeric_only=True),annot=True,cmap='Blues',fmt='.1g')
```

- Relaciona **Sales**, **Profit**, **Discount**, **Quantity**, **Days_to_ship**, etc.
- Traducido esto a negocio:
  - qué variables están más ligadas al beneficio.
  - qué palancas (precio, descuento, volumen, tiempos) parecen más sensibles.

## 6.- LOGÍSTICA Y EXPERIENCIA DE CLIENTE: Ship Mode & Days_to_ship

Se puede distinguir una parte clara de “operaciones”:

- Conteo por **Ship Mode**.
- Ventas y beneficios por modo de envío.
- **Days_to_ship**:
  - distribución general.
  - boxplot de días hasta envío por modo de envío.

La lógica de negocio aquí nos plantearía este análisis entre otros:
“Quiero ver si los modos de envío que ofrezco están alineados con la promesa de servicio y si el coste de la logística tiene relación con la rentabilidad.”

También se puede relacionar el modo de envío con ventas y con margen, lo que permite debatir de:

- posible **trade-off coste de transporte vs satisfacción del cliente**.
- qué modos de envío conviene impulsar o encarecer.

## 7.- SEGMENTOS DE CLIENTES Y MIX COMERCIAL

Se analizan:

- Segment (Consumer, Corporate, Home Office, etc.) por:
  - volumen de registros.
  - ventas.
  - beneficio.

- Cruces Segment por Category y Segment por Region con gráficos de barras.

Conclusión de la lógica de negocio: entender qué tipo de cliente es más rentable y cúal no y qué combinaciones segmento-producto-región son más interesantes para orientar campañas, descuentos y esfuerzos comerciales.

## 8.- PREPARACIÓN PARA ML: DATASET LISTO PARA MODELAR EL PROFIT

La parte final es ingeniería de características para ML (Machine Learning):

**8.1- Copia del dataframe:**

```
# Copia para no perder el dataframe original.
data=df.copy()
```

**8.2- Selección de columnas categóricas a codificar:**

```
cat_cols = []
for c in ['Category','Sub-Category','Segment','Region']:
  if c in data.columns:
    cat_cols.append(c)
```

**8.3- One-hot encoding:**

```
data = pd.get_dummies(data, columns=cat_cols, drop_first=True)
```

- Convierte categorías a variables binarias para consumo por modelos.
- ***drop_first=True*** para evitar multicolinealidad.

**8.4- Filtrado a columnas numéricas:**

```
data=data.select_dtypes(include=['number'])
```

**8.5- Definición de objetivo y features:**

```
y=data['Profit']
X=data.drop(columns=['Profit'])
```

**8.6- Tratamiento simple de nulos:**

```
X=X.fillna(X.median())
```

**8.7- Train / test split con scikit-learn:**

```
X_train,X_test,y_train,y_test=train_test_split(
    X,y,
    test_size=0.2,
    random_state=42
)
```

Aunque en este modelo no se entrena ningún modelo, la lógica de negocio es:
“He dejado el dataset transformado y preparado para construir un modelo que haga **predicciones del beneficio de cada pedido** a partir de variables de producto, cliente, tiempo, descuento y logística. Ese futuro modelo ayudaría para simular escenarios y apoyar decisiones de pricing, promociones y servicio.”

## 9.- RESUMEN DEL RETO

“ En este reto he trabajado como si fuera el P&L de una tienda. Lo primero que hago es validar la calidad de los datos y creo indicadores de negocio clave: coste, margen de beneficio, rentabilidad binaria, tiempo de envío, precio unitario, bandas de descuento y variables temporales.
El siguiente paso es realizar un EDA cruzando estos indicadores por producto, cliente, región, segmento y tiempo para identificar en dónde se gana y en dónde se pierde dinero, y como influyen los descuentos y la logística en el margen de beneficio.
Finalmente, dejo el dataset preparado y transformado con one-hot encoding y train/test split para poder entrenar modelos que predigan el Profit (beneficio) y así usarlo como palanca de decisión en pricing, promociones y gestión de envíos.”

## 10.- VOCABULARIO APRENDIDO

- **Datos nulos:** son valores desconocidos, indefinidoso que no existen en ningún registro.Pueden causar problemas en el análisis, se tratan identificándolos, eliminándo las filas y columnas que los contienen o rellenando estos valores nulos por un valor constante, promedio, mediana o la moda.
- **Precio unitario efectivo:** coste final de cada producto, ya calculado después de aplicar todos los factores relevantes, como impuestos, descuentos, costos de producción, gastos indirectos y margen de beneficio. Refleja el coste total para el comprador.
- **Estacionalidad mensual:** es el patrón predecible de variaciones de los datos que se repite cada mes, es decir las fluctuaciones que ocurren dentro de un año y que son distintas de los ciclos más largos, por ejemplo los meses de vacaciones.
- **Segmentación de pricing:** estrategia de fijación de precios que consiste en vender un mismo producto o servicio a diferentes precios para distintos grupos de clientes, en función de su disposición a pagar, su comportamiento o sus necesidades.
- **Variables transaccionales:** datos dinámicos que describen detalles de una transacción como importe de un carrito de compras o el estado de un envío.
- **One-hot encoding:** técnica de codificación que transforma datos categóricos en un formato númerico que puedan usar los modelos de ML. Para cada categoría única en su columna, crea una nueva columna binaria(0 ó 1).:
  - El valor es 1 si la observación pertenece a esa categoría
  - El valor es 0 si no.
- **P&L (Profit and Loss):** es un estado financiero que resume sus ingresos y gastos durante un período específico para determinar si ha generado beneficios o pérdidas.
- **Multicolinealidad:** alta correlación entre dos o más variables independientes en un modelo de regresión lineal. Dificulta la medición de la relación entre variables, aunque no afecta al modelo si puede hacer que los coeficientes de regresión sean menos fiables.