02_modelo_sQ

November 11, 2025

1 Introducción y Contexto

2 02_Modelo_sQ – Sistema de Revisión Continua

Este notebook implementa el modelo prescriptivo (s, Q) para el caso Retail, usando los datos preparados en el `01_eda.ipynb`.

2.1 Objetivos:

- Calcular el lote económico de pedido (Q*) y el punto de reorden (s).
- Evaluar políticas logísticas bajo distintos escenarios.
- Estimar costos totales (pedido, mantenimiento, rotura y almacenamiento).
- Preparar la base para el Dashboard Prescriptivo.

3 Cargar datos preparados

[6]:
```
import pandas as pd

# Cargar métricas del EDA
metricas_venta = pd.read_csv("metricas_venta_integradas.csv")

# Cargar escenarios logísticos
costos = pd.read_excel("../data/Costos_Logísticos.xlsx")

print("  Datos cargados correctamente")
print("Métricas:", metricas_venta.shape)
print("Escenarios:", costos.shape)
```

```
  Datos cargados correctamente
Métricas: (600, 15)
Escenarios: (5, 7)
```

4 Modelo (s, Q) Base

Q* → lote óptimo de pedido

s → punto de reorden

CTA → costo total anual estimado

[8]:
```
print(costos.columns.tolist())
costos.head()
```

```
['Escenario', 'costo_pedido', 'costo_mantenimiento_anual',
'costo_almacenamiento_m2', 'costo_transferencia_tienda', 'costo_rotura_stock',
'costo_obsolescencia']
```

[8]:

	Escenario	costo_pedido	costo_mantenimiento_anual	costo_almacenamiento_m2	costo_transferencia_tienda	costo_rotura_stock	costo_obsolescencia
0	Escenario 1	10	0.15	80	3	0.20	0.09
1	Escenario 2	40	0.20	100	4	0.25	0.12
2	Escenario 3	50	0.25	120	5	0.30	0.15
3	Escenario 4	55	0.30	130	6	0.35	0.18
4	Escenario 5	60	0.35	140	7	0.38	0.20

[9]:
```
costos.columns = costos.columns.str.strip().str.lower()
print(costos.columns.tolist())
```

```
['escenario', 'costo_pedido', 'costo_mantenimiento_anual',
'costo_almacenamiento_m2', 'costo_transferencia_tienda', 'costo_rotura_stock',
'costo_obsolescencia']
```

[21]:
```
import numpy as np
import pandas as pd

# ------------------------------
# 1  Bucle principal: cálculo (s, Q) + CTE
# ------------------------------
resultados = []

for _, row in costos.iterrows():
    escenario = row['escenario']
    S = row['costo_pedido']
    h = row['costo_mantenimiento_anual']
    Z = 1.65  # Nivel de servicio 95 %
```

```
    print(f"  Procesando {escenario} | S={S} | h={h}")

    df = metricas_venta.copy()
    df['D'] = df['demanda_promedio'] * 52
    df['H'] = h * df['costo_unitario']

    # Modelo (s, Q)
    df['Q_opt'] = np.sqrt((2 * df['D'] * S) / df['H'])
    df['D_L'] = df['demanda_promedio'] * df['lead_time']
    df['sigma_L'] = df['desviacion_demanda'] * np.sqrt(df['lead_time'])
    df['s_reorder'] = df['D_L'] + Z * df['sigma_L']

    # Costo total anual básico
    df['CTA'] = (df['D'] / df['Q_opt']) * S + (df['Q_opt'] / 2) * df['H']

    # Costo total extendido (con tu diccionario)
    df['CTE'] = (
        df['CTA']
        + row['costo_almacenamiento_m2'] * 12 * df['D'] / 52 * 0.05
        + row['costo_transferencia_tienda'] * 0.02 * df['D']
        + row['costo_rotura_stock'] * df['margen_promedio'] * 0.05
        + row['costo_obsolescencia'] * df['margen_promedio'] * 0.03
    )

    df['escenario'] = escenario
    resultados.append(df)

# ----------------------------
# 2 Unir todos los escenarios
# ----------------------------
df_sQ = pd.concat(resultados, ignore_index=True)
print("  Cálculo (s, Q) y CTE completado correctamente.")
```

```
 Procesando Escenario 1 | S=10 | h=0.15
 Procesando Escenario 2 | S=40 | h=0.2
 Procesando Escenario 3 | S=50 | h=0.25
 Procesando Escenario 4 | S=55 | h=0.3
 Procesando Escenario 5 | S=60 | h=0.35
 Cálculo (s, Q) y CTE completado correctamente.
```

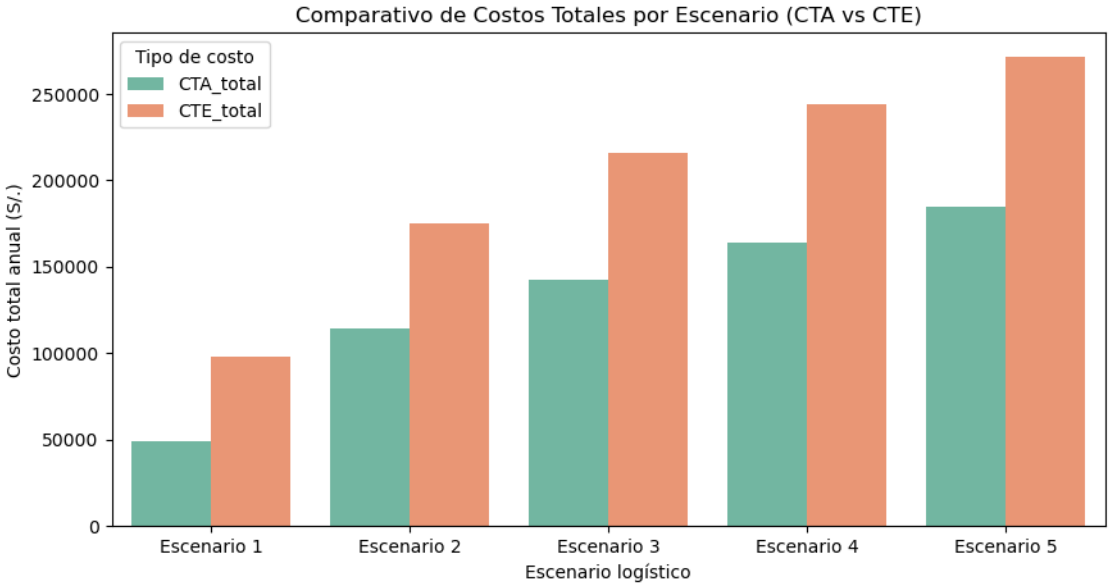
[22]:
```
# ----------------------------
# 3 Resumen comparativo
# ----------------------------
resumen_cte = (
    df_sQ.groupby('escenario')
    .agg(
```

```
        Q_promedio=('Q_opt', 'mean'),
        s_promedio=('s_reorder', 'mean'),
        CTA_total=('CTA', 'sum'),
        CTE_total=('CTE', 'sum')
    )
    .reset_index()
)

print(" Resumen comparativo:")
print(resumen_cte)
```

```
 Resumen comparativo:
     escenario  Q_promedio  s_promedio      CTA_total      CTE_total
0  Escenario 1   21.581585   27.638714   49375.836668   98116.319589
1  Escenario 2   37.380401   27.638714  114028.610366  175204.814764
2  Escenario 3   37.380401   27.638714  142535.762958  216147.688832
3  Escenario 4   35.789009   27.638714  163761.123937  244113.176767
4  Escenario 5   34.607527   27.638714  184747.463996  271817.736469
```

5 Visualización comparativa de escenarios

[23]:
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10,5))
sns.barplot(
    data=resumen_cte.melt(
        id_vars='escenario',
        value_vars=['CTA_total','CTE_total'],
        var_name='TipoCosto', value_name='CostoTotal'
    ),
    x='escenario', y='CostoTotal', hue='TipoCosto', palette='Set2'
)
plt.title("Comparativo de Costos Totales por Escenario (CTA vs CTE)")
plt.xlabel("Escenario logístico")
plt.ylabel("Costo total anual (S/.)")
plt.legend(title='Tipo de costo')
plt.show()
```



[24]:
```
df_sQ.to_csv("resultados_modelo_sQ.csv", index=False)
resumen_cte.to_csv("resumen_escenarios.csv", index=False)
```

[ ]:
```
```