Proyecto de Análisis de Datos – Transactions Dataset

1. Introducción

Este proyecto tiene como objetivo realizar un **análisis exploratorio de datos (EDA)** del dataset `transactions.csv`, el cual contiene información detallada sobre compras realizadas por clientes en diferentes categorías de productos.

El propósito principal es identificar **patrones de consumo, comportamiento de compra, impacto de los descuentos** y **distribución de métodos de pago.**

Este análisis forma parte de mis primeros proyectos prácticos en análisis de datos, utilizando **Python** y librerías como **Pandas.**

En esta primera versión **no se generan gráficos**, centrándonos únicamente en estadísticas descriptivas y conclusiones basadas en los datos.

---

2. Descripción del Dataset

El dataset `transactions.csv` contiene información transaccional donde cada fila representa una compra. Incluye datos demográficos, información del producto, detalles del descuento y montos económicos.

Columnas del dataset:

- **CID:** Identificador único del cliente.
- **TID:** Identificador único de la transacción.
- **Gender:** Género del cliente.
- **Age Group:** Grupo de edad del cliente.
- **Purchase Date:** Fecha y hora de la compra.
- **Product Category:** Categoría del producto adquirido.
- **Discount Availed:** Indica si se aplicó un descuento (Yes/No).
- **Discount Name:** Nombre del descuento.
- **Discount Amount (INR):** Valor del descuento.
- **Gross Amount:** Monto antes del descuento.
- **Net Amount:** Monto pagado tras el descuento.
- **Purchase Method:** Método de pago utilizado.
- **Location:** Ciudad de la compra.

---

3. Objetivos del Análisis

Los objetivos principales son:

Comprensión del cliente

- Analizar **género**, **grupo de edad** y **ciudades principales.**

Análisis del consumo

- Determinar qué **categorías de productos** son más compradas.
- Identificar los **métodos de pago** más utilizados.

Impacto de descuentos

- Medir cuánto se reduce el monto al aplicar promociones.
- Ver qué **descuentos** se usan con mayor frecuencia.

Ingresos

- Calcular ingresos **brutos y netos por categoría**.
- Identificar qué **ciudades generan más ingresos.**

Cargando el dataset

Cargando el dataset desde Kaggle y convirtiendo la columna de fecha a formato datetime.
Esto nos permitirá realizar análisis por día, mes y año de forma correcta.

In [ ]:
```
!pip install pandas
```

In [ ]:
```
!pip install kagglehub
```

In [31]:
```
# Install dependencies as needed:
# pip install kagglehub[pandas-datasets]
import kagglehub
from kagglehub import KaggleDatasetAdapter

# Set the path to the file you'd like to load
file_path = "project1_df.csv"

# Load the latest version
df = kagglehub.load_dataset(
  KaggleDatasetAdapter.PANDAS,
  "shrishtimanja/ecommerce-dataset-for-data-analysis",
  file_path,
  # Provide any additional arguments like
  # sql_query or pandas_kwargs. See the
  # documenation for more information:
  # https://github.com/Kaggle/kagglehub/blob/main/README.md#kaggledatasetadapter
)

# Convertir la columna a datetime
df["Purchase Date"] = pd.to_datetime(df["Purchase Date"], errors="coerce")
```

```
print("First 5 records:", df.head())
```

```
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_19072\868200363.py:10: DeprecationWa
rning: Use dataset_load() instead of load_dataset(). load_dataset() will be remov
ed in a future version.
  df = kagglehub.load_dataset(
Downloading from https://www.kaggle.com/api/v1/datasets/download/shrishtimanja/ec
ommerce-dataset-for-data-analysis?dataset_version_number=1&file_name=project1_df.
csv...
100%|
| 6.66M/6.66M [00:01<00:00, 6.90MB/s]
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_19072\868200363.py:21: UserWarning:
Parsing dates in %d/%m/%Y %H:%M:%S format when dayfirst=False (the default) was s
pecified. Pass `dayfirst=True` or specify a format to silence this warning.
  df["Purchase Date"] = pd.to_datetime(df["Purchase Date"], errors="coerce")
First 5 records:       CID         TID  Gender     Age Group       Purchase Date
\
0  943146  5876328741  Female         25-45 2023-08-30 20:27:08
1  180079  1018503182    Male         25-45 2024-02-23 09:33:46
2  337580  3814082218   Other  60 and above 2022-03-06 09:09:50
3  180333  1395204173   Other  60 and above 2020-11-04 04:41:57
4  447553  8009390577    Male         18-25 2022-05-31 17:00:32

   Product Category Discount Availed    Discount Name  Discount Amount (INR)  \
0       Electronics              Yes        FESTIVE50                  64.30
1       Electronics              Yes  SEASONALOFFER21                 175.19
2          Clothing              Yes  SEASONALOFFER21                 211.54
3  Sports & Fitness               No              NaN                   0.00
4  Sports & Fitness              Yes         WELCOME5                 439.92

   Gross Amount   Net Amount Purchase Method   Location
0    725.304000   661.004000     Credit Card  Ahmedabad
1   4638.991875  4463.801875     Credit Card  Bangalore
2   1986.372575  1774.832575     Credit Card      Delhi
3   5695.612650  5695.612650      Debit Card      Delhi
4   2292.651500  1852.731500     Credit Card      Delhi
```

Total de clientes y distribución por género

Distribución de clientes por género.

Esto nos ayuda a entender el perfil demográfico de nuestra base de clientes.

In [20]:
```
total_clientes = df.shape[0]
total_hombres = (df["Gender"] == "Male").sum()
total_mujeres = (df["Gender"] == "Female").sum()
otros_generos = (df["Gender"] == "Other").sum()

print("Total clientes: ", total_clientes)
print("Total hombres: ", total_hombres)
print("Total mujeres: ", total_mujeres)
print("Total otros generos: ", otros_generos)
```

```
Total clientes:  55000
Total hombres:  18096
Total mujeres:  18454
Total otros generos:  18450
```

Grupo de edad y ciudades principales

Identificación de los clientes principales por edad y ubicación.
Esto permite enfocar estrategias comerciales según los grupos de mayor consumo.

In [21]:
```
clientes_por_edad = df["Age Group"].value_counts().idxmax()
ciudad_clientes = df["Location"].value_counts().head()

print("Grupo de edad con mayores compras: ", clientes_por_edad)
print("Las 5 ciudades con más clientes: \n", ciudad_clientes)
```

```
Grupo de edad con mayores compras:  25-45
Las 5 ciudades con más clientes:
 Location
Mumbai       11197
Delhi        10799
Bangalore     8249
Hyderabad     5545
Chennai       4368
Name: count, dtype: int64
```

Métodos de pago más frecuente

Métodos de pago más usado por los clientes.
Esto permite a la empresa priorizar métodos de pago más convenientes.

In [29]:
```
metodo_pago = df["Purchase Method"].value_counts().head(3)
print("Métodos de pago más frecuente: \n", metodo_pago)
```

```
Métodos de pago más frecuente:
 Purchase Method
Credit Card    22096
Debit Card     13809
Net Banking     5485
Name: count, dtype: int64
```

Ventas por día y mes

Identificación de patrones temporales de compra.
Saber los días y meses con más ventas ayuda a planificar promociones y stock.

In [33]:
```
dia_compras = df["Purchase Date"].dt.day.value_counts().head()
mes_compras = df["Purchase Date"].dt.month.value_counts().head()

print("Días más comprados: \n", dia_compras)
print("Meses más comprados: \n", mes_compras)
```

```
Días más comprados:
 Purchase Date
26    1928
22    1866
28    1865
19    1860
24    1853
Name: count, dtype: int64
Meses más comprados:
 Purchase Date
1     4716
10    4693
3     4666
7     4655
4     4649
Name: count, dtype: int64
```

Categorías más vendidas e ingresos

Análisis de productos más vendidos y que generan mayores ingresos.
Esto permite identificar productos estratégicos y oportunidades de crecimiento.

In [38]:
```
categoria_compras = df["Product Category"].value_counts().head()
categoria_ingresos = df.groupby("Product Category")["Net Amount"].sum().sort_val

print("Categorias con más ventas: \n", categoria_compras)
print("Categorias con mayores ingresos: \n", categoria_ingresos)
```

```
Categorias con más ventas:
 Product Category
Electronics          16574
Clothing             10968
Beauty and Health     8332
Sports & Fitness      5557
Home & Kitchen        5489
Name: count, dtype: int64
Categorias con mayores ingresos:
 Product Category
Electronics          4.748257e+07
Clothing             3.122038e+07
Beauty and Health    2.418552e+07
Sports & Fitness     1.613983e+07
Home & Kitchen       1.589259e+07
Books                7.932802e+06
Other                6.209626e+06
Pet Care             4.637088e+06
Toys & Games         4.476831e+06
Name: Net Amount, dtype: float64
```

Uso de descuentos

Impacto y uso de los descuentos.
Analizar los descuentos más populares ayuda a optimizar campañas de promoción.

In [37]:
```
clientes_descuentos = (df["Discount Availed"] == "Yes").sum()
descuento_mas_usado = df["Discount Name"].value_counts().head()

print("Clientes que usan descuentos: \n", clientes_descuentos)
print("Descuentos más usados: \n", descuento_mas_usado)
```

```
Clientes que usan descuentos:
 27415
Descuentos más usados:
 Discount Name
NEWYEARS           8135
SEASONALOFFER21    6940
FESTIVE50          4115
SAVE10             4115
WELCOME5           4110
Name: count, dtype: int64
```

Comparación Gross vs Net Amount

Comparación de montos brutos vs netos por categoría.

Esto muestra cuánto afectan los descuentos a los ingresos.

In [39]:
```
variacion_precio_descuento = (
    df.groupby("Product Category")[["Net Amount", "Gross Amount"]].sum().round(2
)
variacion_precio_descuento["Reducción (%)"] = (
    (variacion_precio_descuento["Gross Amount"] - variacion_precio_descuento["Ne
    / variacion_precio_descuento["Gross Amount"]
) * 100

print(variacion_precio_descuento)
```

```
                    Net Amount  Gross Amount  Reducción (%)
Product Category
Beauty and Health  24185519.69   25320129.06       4.481057
Books               7932802.02    8312144.30       4.563711
Clothing           31220376.71   32732062.43       4.618364
Electronics        47482567.70   49743506.51       4.545194
Home & Kitchen     15892593.91   16653800.55       4.570768
Other               6209625.94    6512450.46       4.649932
Pet Care            4637087.68    4854486.13       4.478300
Sports & Fitness   16139834.54   16892820.74       4.457433
Toys & Games        4476831.36    4690113.14       4.547476
```

Total ventas y por ciudad

Ingresos totales y por ciudad.

Permite identificar las ciudades que generan mayor facturación.

In [41]:
```
total_ventas = df["Net Amount"].sum()
ingresos_ciudad = df.groupby("Location")["Net Amount"].sum()

print("Total ventas:", total_ventas)
print("Ingresos por ciudad: \n", ingresos_ciudad)
```

```
Total ventas: 158177239.558525
Ingresos por ciudad:
 Location
Ahmedabad    8.114984e+06
Bangalore    2.361955e+07
Chennai      1.263518e+07
Dehradun     1.569219e+06
Delhi        3.109897e+07
Hyderabad    1.581339e+07
Jaipur       4.854492e+06
Kolkata      7.805355e+06
Lucknow      3.270708e+06
Mumbai       3.208384e+07
Other        3.079422e+06
Pune         1.090214e+07
Srinagar     1.585927e+06
Varanasi     1.744066e+06
Name: Net Amount, dtype: float64
```

Conclusiones del Análisis de Datos

Perfil del Cliente

- **Distribución de género equilibrada**: Hombres (18,096), Mujeres (18,454) y Otros géneros (18,450) muestran una distribución casi igualitaria.
- **Grupo de edad principal**: Clientes entre 25-45 años realizan la mayor cantidad de compras.
- **Ciudades más activas**: Mumbai (11,197), Delhi (10,799) y Bangalore (8,249) concentran la mayor base de clientes.

Comportamiento de Compra

- **Método de pago preferido**: Tarjeta de Crédito (22,096 transacciones) es el más utilizado, seguido de Tarjeta de Débito (13,809).
- **Patrones temporales**: Los días 26, 22 y 28 del mes registran mayor actividad de compras. Enero (mes 1) y Octubre (mes 10) son los meses con más ventas.

Desempeño Comercial

- **Categoría más vendida**: Electrónicos (16,574 ventas) lidera en volumen de transacciones.
- **Mayores ingresos**: Electrónicos genera ₹47.48M, seguido de Ropa (₹31.22M) y Belleza & Salud (₹24.19M).
- **Impacto de descuentos**: 27,415 clientes utilizan descuentos, siendo "NEWYEARS" (8,135 usos) el más popular.

Impacto Financiero

- **Reducción por descuentos**: Los descuentos representan aproximadamente 4.5-4.6% de reducción en los ingresos brutos por categoría.
- **Ciudades más rentables**: Delhi (₹31.10M), Mumbai (₹32.08M) y Bangalore (₹23.62M) generan los mayores ingresos netos.
- **Ventas totales**: El volumen total de ventas netas asciende a ₹158.18 millones.

In [ ]: