Data Cleaning Project — (Nike_Sales_Uncleaned.csv)

In [ ]:
```
## Objective - The goal of this project is to clean raw data and prepare it for fur
# This includes:
# Handling missing values
# Cleaning and standardizing numeric fields
# Processing and encoding categorical values
# Parsing and formatting date/time values
# Exporting the final cleaned dataset
```

In [45]:
```
# Import Required Libraries
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
```

In [47]:
```
# Load the Dataset
data=pd.read_csv("Nike_Sales_Uncleaned.csv")
data.head()
```

Out[47]:

	Order_ID	Gender_Category	Product_Line	Product_Name	Size	Units_Sold	MRP	Dis
0	2000	Kids	Training	SuperRep Go	M	NaN	NaN	
1	2001	Women	Soccer	Tiempo Legend	M	3.0	4957.93	
2	2002	Women	Soccer	Premier III	M	4.0	NaN	
3	2003	Kids	Lifestyle	Blazer Mid	L	NaN	9673.57	
4	2004	Kids	Running	React Infinity	XL	NaN	NaN	

Inspect the Data Structure

In [51]:
```
data.shape
```

Out[51]:
```
(2500, 13)
```

In [5]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2500 entries, 0 to 2499
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Order_ID          2500 non-null   int64  
 1   Gender_Category   2500 non-null   object 
 2   Product_Line      2500 non-null   object 
 3   Product_Name      2500 non-null   object 
 4   Size              1990 non-null   object 
 5   Units_Sold        1265 non-null   float64
 6   MRP               1246 non-null   float64
 7   Discount_Applied  832 non-null    float64
 8   Revenue           2500 non-null   float64
 9   Order_Date        1884 non-null   object 
 10  Sales_Channel     2500 non-null   object 
 11  Region            2500 non-null   object 
 12  Profit            2500 non-null   float64
dtypes: float64(5), int64(1), object(7)
memory usage: 254.0+ KB
```

In [6]:
```
data.isnull().sum()
```

Out[6]:
```
Order_ID              0
Gender_Category       0
Product_Line          0
Product_Name          0
Size                510
Units_Sold         1235
MRP                1254
Discount_Applied   1668
Revenue               0
Order_Date          616
Sales_Channel         0
Region                0
Profit                0
dtype: int64
```

Cleaning numerical columns

In [8]:
```
data["Units_Sold"].fillna(data["Units_Sold"].mean(),inplace=True)
data["Units_Sold"].isnull().sum()
```

Out[8]:
```
0
```

In [9]:
```
data["MRP"].fillna(data["MRP"].mean(),inplace=True)
data["MRP"].isnull().sum()
```

Out[9]:
```
0
```

In [10]:
```
data["Discount_Applied"].fillna(data["Discount_Applied"].mean(),inplace=True)
data["Discount_Applied"].isnull().sum()
```

Out[10]:
```
0
```

In [11]:
```
data.isnull().sum()
```

Out[11]:
```
Order_ID              0
Gender_Category       0
Product_Line          0
Product_Name          0
Size                510
Units_Sold            0
MRP                   0
Discount_Applied      0
Revenue               0
Order_Date          616
Sales_Channel         0
Region                0
Profit                0
dtype: int64
```

Cleaning categorical column

In [13]:
```
data["Size"].fillna(data["Size"].mode()[0],inplace=True)
data["Size"].isnull().sum()
```

Out[13]:
```
0
```

In [14]:
```
data.isnull().sum()
```

Out[14]:
```
Order_ID              0
Gender_Category       0
Product_Line          0
Product_Name          0
Size                  0
Units_Sold            0
MRP                   0
Discount_Applied      0
Revenue               0
Order_Date          616
Sales_Channel         0
Region                0
Profit                0
dtype: int64
```

In [15]:
```
data["Size_Type"] = data["Size"].apply(lambda x: "NUMERIC" if x.isdigit() else "ALP
```

In [16]:
```
data.tail()
```

Out[16]:

	Order_ID	Gender_Category	Product_Line	Product_Name	Size	Units_Sold	M
2495	4495	Kids	Basketball	Kyrie Flytrap	XL	3.000000	6039.863
2496	4496	Men	Basketball	Kyrie Flytrap	L	-1.000000	6039.863
2497	4497	Men	Soccer	Tiempo Legend	7	1.482213	6647.600
2498	4498	Women	Training	ZoomX Invincible	L	4.000000	5358.700
2499	4499	Women	Running	Air Zoom	M	1.482213	5550.990

Cleaning Date column

In [18]:
```
data["Order_Date"].value_counts()
```

Out[18]:
```
Order_Date
17-11-2024    6
2024/12/16    6
19-07-2025    6
2024/11/10    6
10-12-2024    6
             ..
2023-10-26    1
2024-08-25    1
2025/07/15    1
2023-09-26    1
2025-05-14    1
Name: count, Length: 1008, dtype: int64
```

In [19]:
```
data["Order_Date"]= pd.to_datetime(data["Order_Date"], errors="coerce")
data["Order_Date"] = data["Order_Date"].dt.strftime("%Y-%m-%d")
data["Order_Date"] = data["Order_Date"].fillna("UNKNOWN")
data["Order_Date"]
```

Out[19]:
```
0       2024-03-09
1       2024-07-09
2          UNKNOWN
3          UNKNOWN
4          UNKNOWN
             ...
2495    2025-05-14
2496       UNKNOWN
2497       UNKNOWN
2498       UNKNOWN
2499       UNKNOWN
Name: Order_Date, Length: 2500, dtype: object
```

In [20]:
```
data.head()
```

Out[20]:

	Order_ID	Gender_Category	Product_Line	Product_Name	Size	Units_Sold	MRP
0	2000	Kids	Training	SuperRep Go	M	1.482213	6039.863395
1	2001	Women	Soccer	Tiempo Legend	M	3.000000	4957.930000
2	2002	Women	Soccer	Premier III	M	4.000000	6039.863395
3	2003	Kids	Lifestyle	Blazer Mid	L	1.482213	9673.570000
4	2004	Kids	Running	React Infinity	XL	1.482213	6039.863395

Export Final Cleaned Dataset

In [21]:
```
data.to_csv("Cleaned_data.csv", index=False)
```

In [42]:
```
data.to_excel("Cleaned_data.xlsx", index=False)
```

Conclusion :

The dataset has been successfully cleaned:

All numeric, categorical, and date fields are standardized.

No missing or invalid values remain.

Dataset is now ready for feature engineering, visualization, or machine learning.

In [ ]: