In [80]:
```
import pandas as pd

# Load the dataset
file_path = "C:/Users/gundr/Downloads/Day 19_E-Commerce_Data.csv"
df = pd.read_csv(file_path)

# Display basic information and first few rows
df.info(), df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 505 entries, 0 to 504
Data columns (total 7 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Order_ID          505 non-null    int64  
 1   Product_Category  481 non-null    object 
 2   Product_Price     485 non-null    float64
 3   Customer_Age      490 non-null    float64
 4   Rating            480 non-null    float64
 5   Review_Text       484 non-null    object 
 6   Order_Date        495 non-null    object 
dtypes: float64(3), int64(1), object(3)
memory usage: 27.7+ KB
```

Out[80]:
```
(None,
    Order_ID Product_Category  Product_Price  Customer_Age  Rating  \
  0        1         Clothing         3262.0          58.0     3.0   
  1        2         Clothing          214.0          20.0     3.0   
  2        3       Home Decor         3429.0          51.0     2.0   
  3        4            Books         4568.0          35.0     3.0   
  4        5      Electronics         2237.0          32.0     4.0   

             Review_Text  Order_Date  
  0  Would not recommend  2023-01-01  
  1   Excellent product!  2023-01-02  
  2      Value for money  2023-01-03  
  3      Value for money  2023-01-04  
  4      Not as expected  2023-01-05  )
```

In [141…
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.impute import SimpleImputer
```

In [119…
```
# Load the dataset
file_path = "C:/Users/gundr/Downloads/Day 19_E-Commerce_Data.csv"
df = pd.read_csv(file_path)
```

In [121…
```
# Identify missing values
missing_values = df.isnull().sum()
print("Missing Values:\n", missing_values)
```

```
Missing Values:
 Order_ID            0
Product_Category    24
Product_Price       20
Customer_Age        15
Rating              25
Review_Text         21
Order_Date          10
dtype: int64
```

In [123…
```
# Impute missing numerical values using median
num_imputer = SimpleImputer(strategy="median")
df["Customer_Age"] = num_imputer.fit_transform(df[["Customer_Age"]])
df["Rating"] = num_imputer.fit_transform(df[["Rating"]])
df["Product_Price"] = num_imputer.fit_transform(df[["Product_Price"]])
```

In [125…
```
# Handling missing textual data using NLP (filling with 'No Review')
def fill_missing_reviews(text):
    if pd.isnull(text) or text.strip() == "":
        return "No Review"
    return text

df["Review_Text"] = df["Review_Text"].apply(fill_missing_reviews)
```

In [127…
```
# Detect and remove duplicate reviews
df.drop_duplicates(subset=["Review_Text"], keep="first", inplace=True)

# Standardize Rating values (ensure they range between 1-5)
df["Rating"] = df["Rating"].clip(1, 5)
```

In [ ]:
```
# Detecting and handling outliers using boxplots
fig, axes = plt.subplots(1, 2, figsize=(12, 5))
sns.boxplot(y=df["Product_Price"], ax=axes[0])
axes[0].set_title("Product Price Outliers")
sns.boxplot(y=df["Rating"], ax=axes[1])
axes[1].set_title("Rating Outliers")
plt.show()
```

In [ ]:
```
# Handling outliers using IQR method
Q1 = df["Product_Price"].quantile(0.25)
Q3 = df["Product_Price"].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
df["Product_Price"] = np.where(df["Product_Price"] < lower_bound, lower_bound, df["Product_Price"])
df["Product_Price"] = np.where(df["Product_Price"] > upper_bound, upper_bound, df["Product_Price"])
```

In [ ]:
```
# Convert categorical data into numerical format (One-hot encoding for Product_Category)
df = pd.get_dummies(df, columns=["Product_Category"], drop_first=True)
```

In [135…
```
# Save cleaned data to a CSV file
cleaned_file_path = "C:/Users/gundr/Downloads/Day 19_E-Commerce_Data.csv"
df.to_csv(cleaned_file_path, index=False)
```

In [137…
```
print("Cleaned dataset saved to:", cleaned_file_path)
```

```
Cleaned dataset saved to: C:/Users/gundr/Downloads/Day 19_E-Commerce_Data.csv
```

In [ ]:
```

```