E-Commerce Product Return Analysis – Data Cleaning & Preparation

Introduction

This project focuses on analyzing and reducing product return rates in an e-commerce business. By understanding the patterns behind returned products, we aim to provide data-driven insights that help the company improve product quality, optimize marketing, and enhance customer satisfaction.

The goal of this step is to clean and prepare the dataset for visualizationg. This involves handling missing values.

Dataset Description

The dataset used in this project is a **synthetic e-commerce returns dataset** sourced from Kaggle.

In [1]:
```
pip install pandas matplotlib seaborn
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pandas in c:\programdata\anaconda3\lib\site-packag
es (2.2.2)
Requirement already satisfied: matplotlib in c:\programdata\anaconda3\lib\site-pa
ckages (3.9.2)
Requirement already satisfied: seaborn in c:\programdata\anaconda3\lib\site-packa
ges (0.13.2)
Requirement already satisfied: numpy>=1.26.0 in c:\programdata\anaconda3\lib\site
-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\programdata\anaconda3
\lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\programdata\anaconda3\lib\site-
packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\programdata\anaconda3\lib\sit
e-packages (from pandas) (2023.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\programdata\anaconda3\lib\s
ite-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\programdata\anaconda3\lib\site-
packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\programdata\anaconda3\lib
\site-packages (from matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\programdata\anaconda3\lib
\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\programdata\anaconda3\lib\si
te-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=8 in c:\programdata\anaconda3\lib\site-pac
kages (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\programdata\anaconda3\lib\s
ite-packages (from matplotlib) (3.1.2)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-pack
ages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

Step 1: Import Dataset into Python

In [2]:
```
import pandas as pd

df = pd.read_csv("ecommerce_returns_synthetic_data.csv")
df.head()
```

Out[2]:

	Order_ID	Product_ID	User_ID	Order_Date	Return_Date	Product_Cate
0	ORD00000000	PROD00000000	USER00000000	2023-08-05	2024-08-26	Clo
1	ORD00000001	PROD00000001	USER00000001	2023-10-09	2023-11-09	B
2	ORD00000002	PROD00000002	USER00000002	2023-05-06	NaN	
3	ORD00000003	PROD00000003	USER00000003	2024-08-29	NaN	
4	ORD00000004	PROD00000004	USER00000004	2023-01-16	NaN	B

Step 2: Clean & Prepare the Data

1. Remove duplicates & nulls

In [5]:
```
df.drop_duplicates(inplace=True)
df.dropna(inplace=True)
```

2. Convert dates

In [9]:
```
df['Order_Date'] = pd.to_datetime(df['Order_Date'])
df['Return_Date'] = pd.to_datetime(df['Return_Date'])
```

3. Create ReturnDays

In [13]:
```
df['Days_to_Return'] = (df['Return_Date'] - df['Order_Date']).dt.days
df['Days_to_Return'] = df['Days_to_Return'].round().astype('Int64')
```

4. Create target column

In [11]:
```
df.rename(columns={
    'Return_Status': 'IsReturned',
    'Return_Date': 'ReturnDate',
    'Days_to_Return': 'ReturnDays'
}, inplace=True)
df.head()
```

Out[11]:

	Order_ID	Product_ID	User_ID	Order_Date	ReturnDate	Product_Categ
0	ORD00000000	PROD00000000	USER00000000	2023-08-05	2024-08-26	Clotl
1	ORD00000001	PROD00000001	USER00000001	2023-10-09	2023-11-09	Bc
5	ORD00000005	PROD00000005	USER00000005	2024-02-14	2024-09-22	Electro
6	ORD00000006	PROD00000006	USER00000006	2023-05-29	2023-08-03	Clotl
7	ORD00000007	PROD00000007	USER00000007	2023-02-09	2024-08-01	Electro

Step 3: Exploratory Data Analysis (EDA)

Analyze Return Rate by:

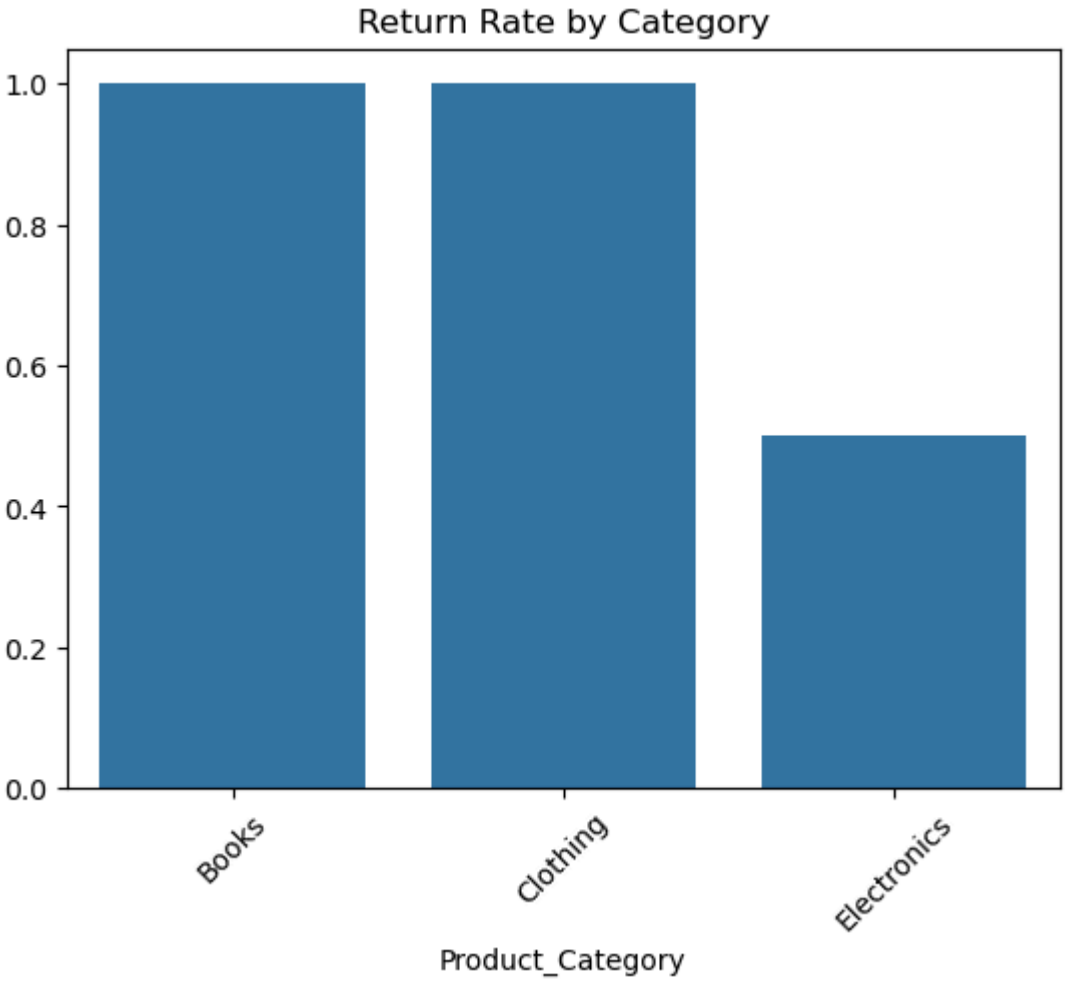
In [26]:
```
# Category-wise return rate
category_return = df.groupby('Product_Category')['IsReturned'].mean()

# Location-wise return rate
location_return = df.groupby('User_Location')['IsReturned'].mean()
```

Visualize:

In [30]:
```
import seaborn as sns
import matplotlib.pyplot as plt

sns.barplot(x=category_return.index, y=category_return.values)
plt.title("Return Rate by Category")
plt.xticks(rotation=45)
plt.show()
```



Step 4: Train Logistic Regression Model

In [39]:
```
features = ['Product_Category', 'Product_Price', 'Discount_Applied', 'User_Locat
X = df_model[features]
y = df_model['IsReturned']
```

In [40]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_
model = LogisticRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
```

```
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[1]]
              precision    recall  f1-score   support

           1       1.00      1.00      1.00         1

    accuracy                           1.00         1
   macro avg       1.00      1.00      1.00         1
weighted avg       1.00      1.00      1.00         1
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:40
9: UserWarning: A single label was found in 'y_true' and 'y_pred'. For the confus
ion matrix to have the correct shape, use the 'labels' parameter to pass all know
n labels.
  warnings.warn(
```

predicted probabilities:

In [41]:
```
df['Return_Probability'] = model.predict_proba(X)[:,1]
df.to_csv("model_output.csv", index=False)
```

Objective of this Notebook

- Clean the raw dataset and ensure data quality
- Save the cleaned dataset for Power BI