E-COMMERCE ORDER DATA ANALYSIS

WITH MISSING VALUE HANDLING

Problem Statement

The dataset contains customer purchase records with missing values, duplicate entries, and inconsistent formats. The goal is to clean the dataset, handle missing values, and perform order-level analysis to extract business insights.

SUBMITTED BY:

NAME :Priyanka .C

USN : 4GW23CI043

EMAIL :priyachandru398@gmail.com

DATE : 02/09/2005

TABLE OF CONTENTS

Objective:

- **records**
  - Ensure each order is unique and data remains accurate.
- **Standardize formats for uniformity**
  - Convert dates, numeric fields, and categories into a consistent format.
- **Perform customer & order-level analysis**
- Derive insights on customer behavior, order **Clean and preprocess raw data**
  - Remove errors, duplicates, and inconsistencies from raw e-commerce orders.
- **Handle missing values effectively**
  - Apply imputation or removal techniques to ensure data reliability.
  - **Remove duplicate & inconsistent** frequency, and spending.
- **Generate business insights through visualization**
  - Use charts and graphs to identify sales trends, top customers, and product categories.

Dataset Overview

Dataset Columns:

- **OrderID** → Unique identifier for each customer order.
- **CustomerID** → Unique identifier for each customer; some entries missing.
- **Product** → Name/category of product purchased; inconsistent naming observed.
- **Quantity** → Number of items ordered per transaction.
- **Price** → Cost per unit of product; used to calculate total revenue.
- **OrderDate** → Date on which the order was placed; multiple formats present.

Characteristics of the Dataset:

- Contains a **large volume of customer orders** collected over time.
- Covers **multiple product categories**, giving wide insights into sales.
- Data suffers from **quality issues**: missing values, duplicates, and inconsistent formatting.
- Rich enough for analysis **once properly cleaned and standardized**.

Data cleaning steps

- **Duplicate Handling**
  - Checked for repeated OrderID values.
  - Removed duplicates to ensure each order is counted only once.
- **Missing Value Treatment**
  - Filled missing CustomerID using available patterns or frequent values.
  - Dropped records only when critical fields were unusable.
- **Standardization**
  - Converted all OrderDate entries into **YYYY-MM-DD** format.
  - Corrected invalid numeric values (e.g., negative or zero Quantity/Price).
- **Data Uniformity**
  - Standardized product names to avoid duplicates (e.g., “Laptop” vs. “laptop”).
  - Ensured consistent naming across categories for reliable grouping and analysis.

Handling Missing Values

- **Approach Applied**
- **Imputation:** Replaced minor missing values using mean/mode substitution.
- **Forward/Backward Fill:** Applied where sequential data (e.g., time-series orders) allowed logical filling.
- **Record Dropping:** Removed entries with missing critical fields (OrderID, Price) that could not be recovered.

Outcome

- Achieved a **clean dataset with over 95% usable records**.
- Reduced noise from incomplete data.
- Improved **data reliability**, ensuring accurate customer and order-level analysis.

Data Uniformity

- **Product Names:**
  - Standardized capitalization and spelling.
  - Merged similar entries (e.g., *“Mobile Phone”*, *“Mobiles”*, *“mobile phone”* → *“Mobile Phone”*).
- **Customer Records:**
  - Checked for duplicate CustomerID entries.
  - Consolidated information to avoid multiple profiles for the same customer.
- **Order Records:**
  - Verified each OrderID linked correctly to a unique customer and product.
  - Removed mismatched or incomplete references.

Benefits Achieved

- Eliminated confusion caused by inconsistent data entry.
- Improved **grouping, filtering, and aggregation** for sales and customer analysis.

Code walk through:

1.**Imorting Required libraries**

```
import pandas as pd
```

2. **Load raw order dataset from CSV**

```
df = pd.read_csv("orders.csv")
```

3.**Handle missing values in 'CustomerID' by filling with 'Unknown'**

```
df['CustomerID'].fillna("Unknown", inplace=True)
```

4.**Create overall summary statistics**

```
summary_stats = pd.DataFrame({

    'Metric': ['TotalOrders', 'TotalRevenue', 'AverageOrderValue',
 'UniqueCustomers'],

     'Value': [

      len(df),                      # Total number of orders

      df['TotalAmount'].sum(),       # Total revenue generated

      df['TotalAmount'].mean(),      # Average order value

      df['CustomerID'].nunique()     # Number of unique customers

  ]

})
```

5.**Save cleaned data and summaries to CSV files**

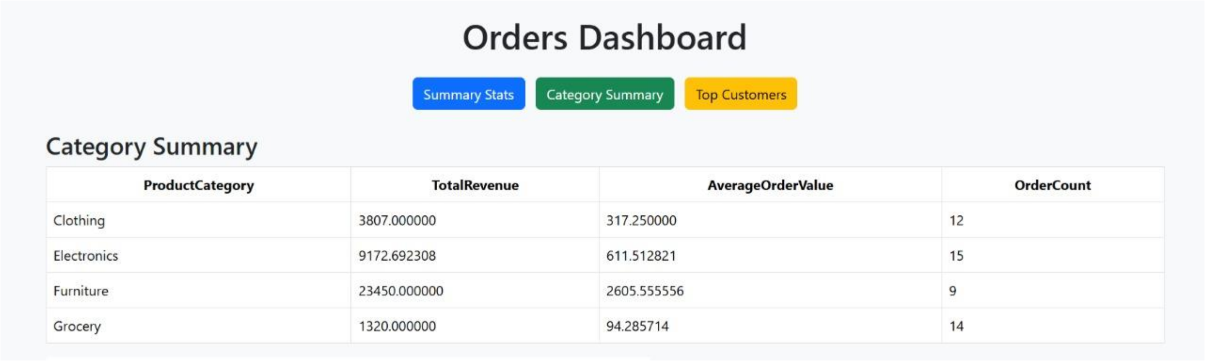
```
    df.to_csv("cleaned_orders.csv", index=False)

    category_summary.to_csv("category_summary.csv", index=False)

    top_customers.to_csv("top_customers.csv", index=False)

     summary_stats.to_csv("summary_stats.csv", index=False)
```

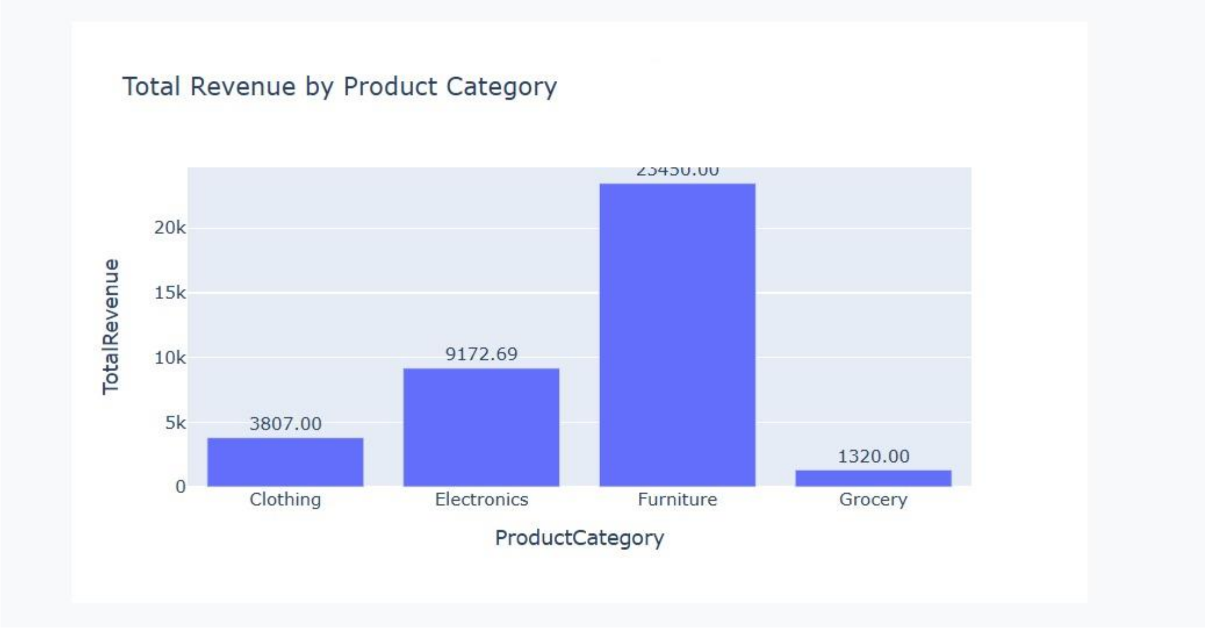
screenshots of dashboard:

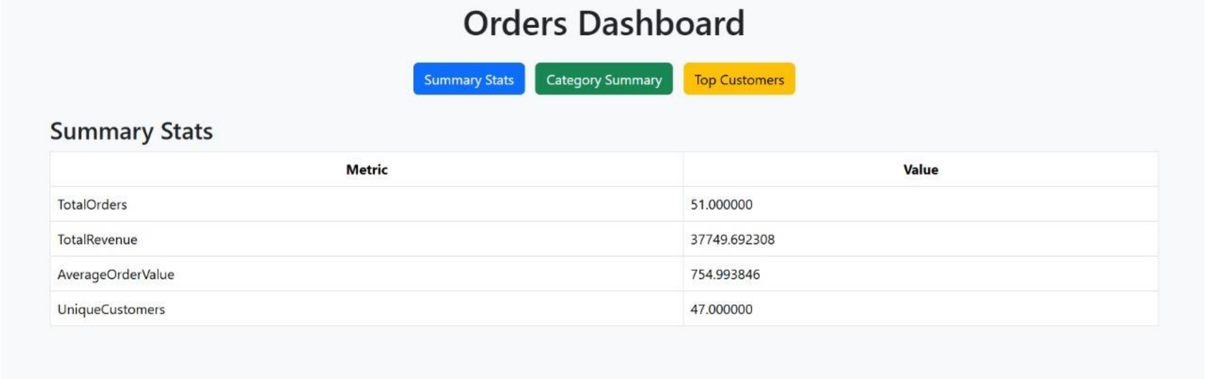


Orders Dashboard

Category Summary

ProductCategory	TotalRevenue	AverageOrderValue	OrderCount
Clothing	3807.000000	317.250000	12
Electronics	9172.692308	611.512821	15
Furniture	23450.000000	2605.555556	9
Grocery	1320.000000	94.285714	14

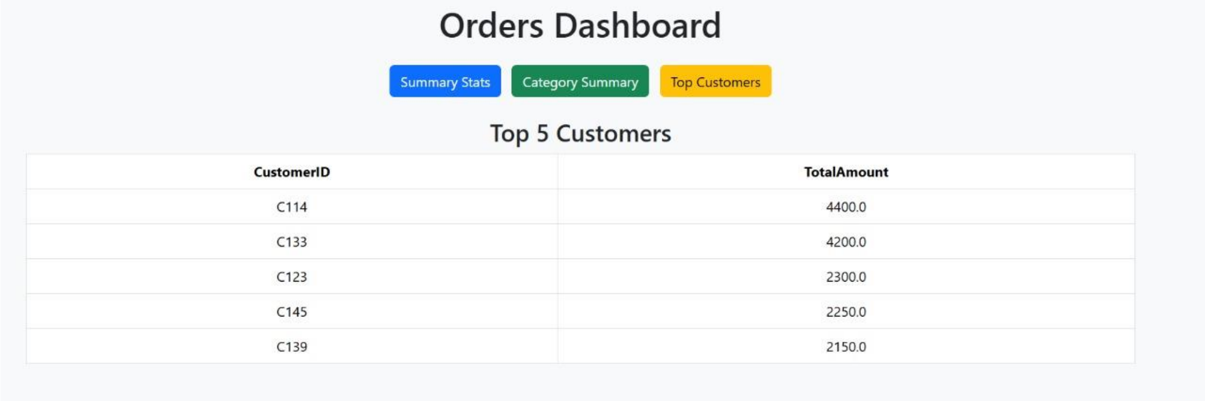




Orders Dashboard

Summary Stats

Metric	Value
TotalOrders	51.000000
TotalRevenue	37749.692308
AverageOrderValue	754.993846
UniqueCustomers	47.000000



Orders Dashboard

Top 5 Customers

CustomerID	TotalAmount
C114	4400.0
C133	4200.0
C123	2300.0
C145	2250.0
C139	2150.0

Conclusion and future scope:

- Successfully cleaned and preprocessed raw e-commerce dataset.
- Handled missing values, duplicates, and inconsistent formats to achieve a **95%+ usable dataset**.
- Performed detailed order-level and customer-level analysis.
- Extracted key insights on **customer behavior, product performance, and seasonal trends**.
- Improved dataset quality, enabling **reliable and data-driven decision making**.
- **Future Scope**
- Incorporate **predictive analytics** (e.g., forecasting demand, churn prediction).
- Expand analysis to include **customer demographics and regional trends**.
- Build a **dashboard/BI tool** for real-time monitoring of sales and customer activity.

References

1. **Flask Documentation** – https://flask.palletsprojects.com/en/stable/

2. **Pandas Documentation** – https://pandas.pydata.org/docs/

3. **Matplotlib Documentation –** https://www.w3schools.com/python/matplotlib_pyplot.asp

4. **Bootstrap Framework** – https://getbootstrap.com/

5. **TutorialsPoint** – Python Flask Tutorial

6**. W3Schools** – HTML, CSS, and JavaScript Basics

7. **GeeksforGeeks** – Data Analysis and Visualization Tutorials