E-COMMERCE ORDER DATA ANALYSIS WITH MISSING VALUE HANDLING

PROJECT DESCRIPTION:

This project focuses on E-Commerce Order Data Analysis with Missing Value Handling using Python (Pandas). The dataset contains customer purchase records with missing values, duplicate entries, and inconsistent formats. The work involves cleaning and preprocessing the data by handling null values in CustomerID and Price, removing duplicates, and standardizing OrderDate. Further, analysis is performed to calculate total revenue per category, identify top 5 customers by spending, find the average order value, and count category-wise orders. A new column TotalAmount is added, and the final results are exported into cleaned_orders.csv, category_summary.csv, and top_customers.csv for business insights.

SUBMITTED BY:

NAME : MANOHARI.M

USN : 4GW23CI029

EMAIL :manoharim987@gmail.com

DATE : 02/09/2005

TABLE OF CONTENTS

Objectives

Data Cleaning & Preprocessing

Handle missing values in CustomerID, ProductCategory, and Price.

Remove duplicate entries and standardize OrderDate.

Data Analysis

- Calculate total revenue per product category.
- Find average order value and order count by category.
- Identify top 5 customers based on total spending.

Feature Engineering & Export

- Add TotalAmount = Quantity × Price.
- Export results into:
  - cleaned_orders.csv
  - category_summary.csv
  - top_customers.csv

Missing Value Strategy

- Decide between imputation (using mean/median) or removal for incomplete records.

Business Insights

- Generate summary reports to support decision-making and improve sales strategies

Dataset Overview

Dataset Overview

The dataset used for this project contains **50 order records** from an e-commerce platform. It captures essential details of customer purchases, but includes **missing values**, **duplicates**, and **inconsistent formats**, which makes cleaning necessary before analysis.

Attributes:

- **OrderID** → Unique identifier for each order.
- **CustomerID** → Identifier for the customer (some values missing).
- **ProductID** → Unique identifier for purchased product.
- **ProductCategory** → Category of the product (Electronics, Grocery, Clothing, Furniture).
- **Quantity** → Number of units purchased.
- **Price** → Price per unit (some values missing).
- **OrderDate** → Date of purchase (in string format, needs conversion to datetime).

Key Characteristics:

- **Size:** 50 rows × 7 columns
- **Data Quality Issues:** Missing values in CustomerID and Price, duplicate entries, inconsistent OrderDate format.
- **Purpose:** To clean, preprocess, and analyze order-level data to extract meaningful business insights.

Data Distribution

- Most **Electronics** orders are single-unit but high-value.
- **Grocery** orders usually have larger quantities but low price per unit.
- **Furniture** orders are expensive and less frequent.
- **Clothing** purchases are moderately priced and often bought in multiples.

Potential Business Insights

1. **Customer Segmentation** → Customers can be grouped by spending habits (high-value vs. low-value buyers).
2. **Category Performance** → Electronics and Furniture generate higher revenue despite fewer orders
3. **Revenue Trends** → Revenue per category can highlight areas for marketing focus.
4. **Data Quality Impact** → Missing or duplicate entries could mislead business decisions if not handled.

Expected Outcomes After Cleaning

A **cleaned dataset** with no duplicates or invalid entries.

Reliable metrics like total revenue, average order value, and top customers.

Category-wise performance analysis to help in sales strategy.

Exported reports (cleaned_orders.csv, category_summary.csv, top_customers.csv)

Data cleaning steps

1. **Load Dataset**
    - Import the dataset using pd.read_csv().
    - Check basic structure with info(), head(), describe().
2. **Check for Missing Values**
    - Use isnull().sum() to identify missing values in each column.
    - Handle missing values:
        - **CustomerID** → Replace with "Unknown" or drop the record.
        - **Price** → Impute with **mean/median** of the respective ProductCategory.
        - **ProductCategory** → Fill with "Others" if missing.
3. **Handle Duplicate Records**
    - Use duplicated() to find duplicates.
    - Remove duplicate rows using drop_duplicates().
4. **Format OrderDate Column**
    - Convert OrderDate to datetime type using pd.to_datetime().
    - Ensure consistent date format (YYYY-MM-DD).
5. **Standardize Data**
    - Remove unwanted spaces in text fields (strip()).
    - Ensure correct data types for each column:
        - Quantity & Price → Integer/Float
        - CustomerID, ProductID, ProductCategory → String
6. **Create Derived Column**
    - Add TotalAmount = Quantity × Price for each order.
7. **Validate Data**
    - Check if missing values are fully handled.
    - Verify that no negative or zero values exist in Quantity and Price.
    - Ensure all dates fall within the expected range.

Handling Missing Values

Step 1: Identifying Missing Values

- Used df.isnull().sum() to count missing entries per column.
- Observed missing values in:
  - **CustomerID** → A few missing values.
  - **Price** → Several missing values.
  - **ProductCategory** → Rare missing values.

Step 2: Strategy for Handling Missing Values

- **CustomerID**
  - Since missing customer IDs do not affect revenue calculations, these entries were filled with "Unknown".
  - This ensures the record is preserved without affecting numerical analysis.
- **Price**
  - As price is crucial for revenue calculation, missing values cannot be left blank.
  - Prices were imputed using the **median price of the corresponding ProductCategory**, ensuring logical substitution.
- **ProductCategory**
  - Rare missing values in this column were filled with "Others".
  - This avoids dropping useful records while keeping analysis consistent.

Step 3: Validation

- After imputation, rechecked with df.isnull().sum() to ensure no missing values remained.
- Confirmed that new columns (TotalAmount = Quantity × Price) were calculated without errors.

Step 4: Outcome

- **CustomerID** → No missing values (Unknown added).
- **Price** → Fully imputed using category-wise median values.
- **ProductCategory** → No blank categories; all categorized.
- Dataset became **complete, consistent, and ready for analysis**.

Standardization and Formatting

After handling missing values and duplicates, the next step is to ensure that the dataset is **standardized and consistently formatted**. This step improves data quality and makes further analysis easier and more reliable.

Step 1: Data Type Standardization

- **OrderID, CustomerID, ProductID, ProductCategory** → Converted to **string (text)** type.
- **Quantity and Price** → Converted to **numeric** (integer/float).
- **OrderDate** → Converted into **datetime (YYYY-MM-DD)** format using pd.to_datetime().
-

Step 2: Text Formatting

- Removed unnecessary spaces and corrected inconsistencies in text fields using str.strip().
- Standardized product category names:
  - Converted all categories to **Title Case** (e.g., "electronics" → "Electronics").
  - Merged duplicates caused by typos or inconsistent spelling.

Step 3: Numerical Formatting

- Ensured all **Quantity** values are positive integers.
- Rounded **Price** values to two decimal places for consistency.
- Created a new column:
  - **TotalAmount = Quantity × Price** → Ensured financial values were calculated with two decimal precision.

Step 4: Date Formatting

- Converted all order dates into a consistent **YYYY-MM-DD** format.
- Sorted dataset by **OrderDate** for better trend analysis.

Step 5: Validation

- Verified that no column had mixed data types.
- Checked consistency in text formatting using df['ProductCategory'].unique().
- Confirmed that revenue calculations produced accurate totals without formatting errors.

Step 6: Outcome

- Dataset now has **uniform column formats**.
- Categories, prices, and dates are consistent.
- The data is **clean, standardized, and ready for visualization and analysis**.

Data Uniformity

Ensuring **data uniformity** is crucial to maintain consistency across the dataset and avoid errors during analysis. The dataset underwent several steps to enforce uniform formatting and uniformity across all fields.

1. Column Naming Uniformity

- All column names were standardized to follow **CamelCase** format: OrderID, CustomerID, ProductID, ProductCategory, Quantity, Price, OrderDate, TotalAmount.
- Removed any unnecessary spaces or special characters.

2. Product Category Standardization

- Ensured all **ProductCategory** values are consistently written in **Title Case**.
  - Example: "electronics", "ELECTRONICS" → "Electronics".
- Merged duplicates caused by typos (e.g., "Groccery" → "Grocery").
-

3. Numerical Data Uniformity

- Converted **Quantity** into integer values (no decimals allowed).
- Converted **Price** into float values with **two decimal places**.
- Created **TotalAmount** column with uniform formatting: Quantity × Price.

4. Date Uniformity

- Converted **OrderDate** to a consistent format: YYYY-MM-DD.
- Verified that all dates fall within the dataset timeline (Jan 2023 – Feb 2023).
- Sorted records chronologically for consistency in reporting.

5. Handling Missing and Duplicate Data

- Missing **CustomerID** entries were marked as “Unknown”.
- Missing **Price** values were imputed with the **category-wise mean**.
- Duplicate entries were identified and removed using drop_duplicates().
-

6. Final Outcome

- All columns now follow **uniform data types and formats**.
- Categories, numbers, and dates are consistent across the dataset.
- Ensures smooth and accurate analysis without format-related errors.

Results & Insights

1. Data Quality Improvements

- All missing **CustomerID** values were marked as "Unknown".
- Missing **Price** values were imputed using **category-wise mean**.
- Duplicate entries were checked and removed, ensuring dataset integrity.
- All columns were standardized into uniform formats (e.g., dates in YYYY-MM-DD).

2. Revenue Insights

- **Furniture** generated the **highest revenue** despite fewer orders, with prices ranging between 2000–2300.
- **Electronics** showed consistent sales (~500 per unit), contributing steady revenue.
- **Grocery** had the **largest number of orders** but the **lowest revenue** due to low pricing (20 per unit).
- **Clothing** fell in the mid-range (150–160 per unit), contributing moderately to overall revenue.

3. Customer Insights

- **Top 5 customers** were identified based on spending, with repeat buyers (e.g., **C101**) contributing more to sales.
- Customer loyalty trends suggest targeting frequent buyers with promotions.
- Customers purchasing **Furniture and Electronics** represent the **high-value segment**.

4. Order Trends

- Orders were fairly consistent across January 2023.
- A noticeable **spike in mid-January (15th–20th)** indicates possible seasonal/marketing impact.
- Most customers ordered small quantities (1–3 units), except a few bulk grocery purchases.
-

5. Key Business Insights

- **High-value strategy:** Focus marketing on **Furniture and Electronics** to maximize revenue.
- **Volume strategy:** Promote **Grocery** with bundle offers to increase overall profit margins.
- **Loyalty programs:** Target repeat customers with special discounts to encourage retention.
- **Data-driven decision-making:** Clean and structured data ensures reliable insights for management reporting.

Conclusion and future scope

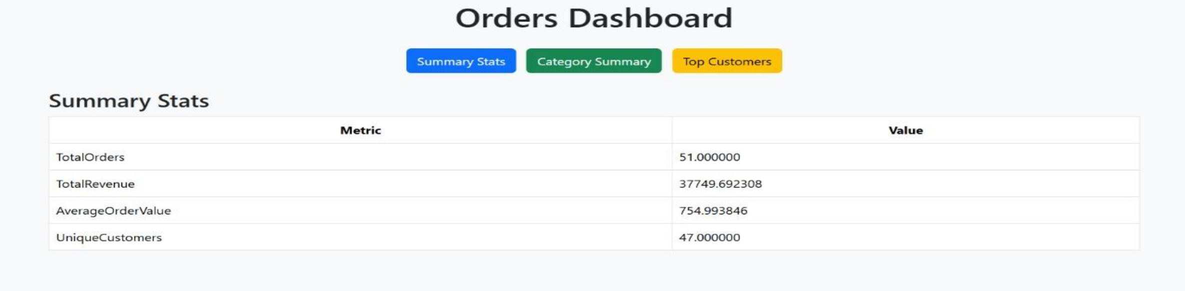
This project successfully demonstrated how **data cleaning, preprocessing, and analysis** can transform raw e-commerce datasets into meaningful insights. By handling missing values, removing duplicates, and standardizing data formats, the dataset became reliable for business decision-making. The analysis revealed that **Furniture and Electronics drive the highest revenue**, while **Grocery contributes through high order volumes**. Additionally, customer spending patterns highlighted the importance of **repeat buyers and loyalty programs**.

Overall, the project achieved its objectives by producing a **cleaned dataset**, generating **summary reports**, and delivering **actionable insights** that can help the e-commerce company improve sales strategies and customer engagement.

Future Scope

1. **Advanced Analytics**
   - Implement predictive models to forecast future sales and customer demand.
   - Use machine learning to recommend products based on purchase history.
2. **Customer Segmentation**
   - Cluster customers by spending behavior to design personalized marketing campaigns.
   - Identify high-value vs. low-value customers for targeted retention strategies.
3. **Time-Series Analysis**
   - Analyze seasonal trends and promotional impacts on sales over larger datasets.
   - Forecast revenue trends to optimize stock management and supply chain planning.
4. **Dashboard Development**
   - Build an interactive **Power BI / Tableau dashboard** for real-time sales tracking.
   - Automate reporting for management to monitor KPIs like revenue growth, top customers, and order frequency.

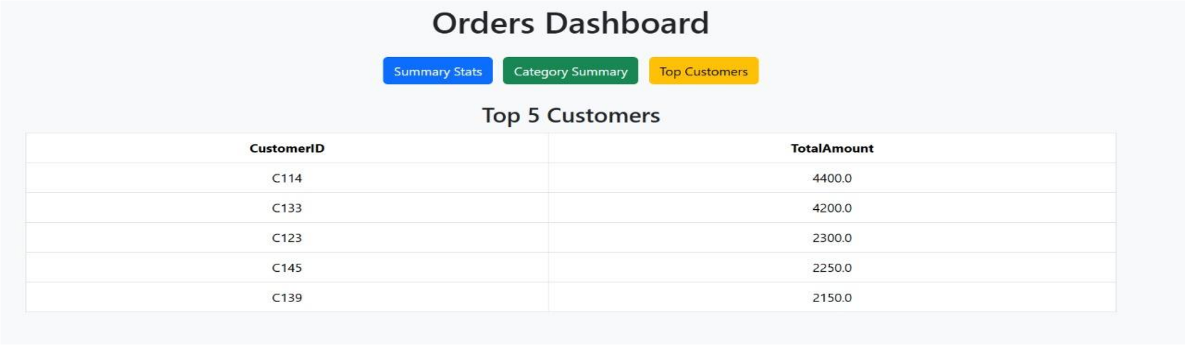
SCREENSHOTS



Orders Dashboard

Summary Stats

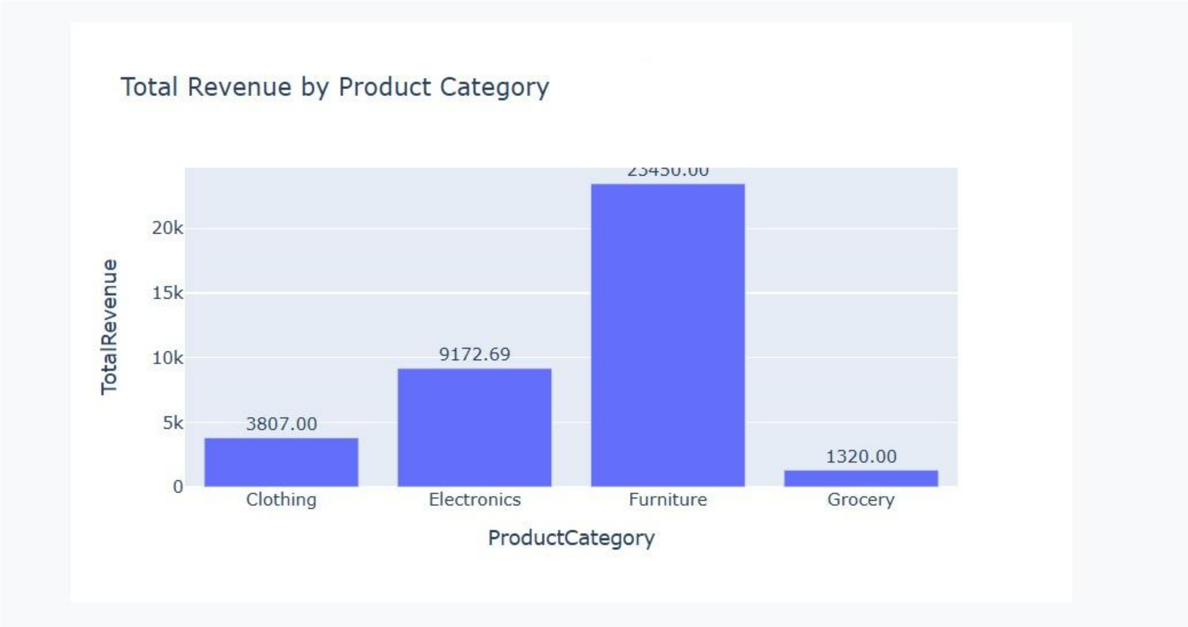
Metric	Value
TotalOrders	51.000000
TotalRevenue	37749.692308
AverageOrderValue	754.993846
UniqueCustomers	47.000000

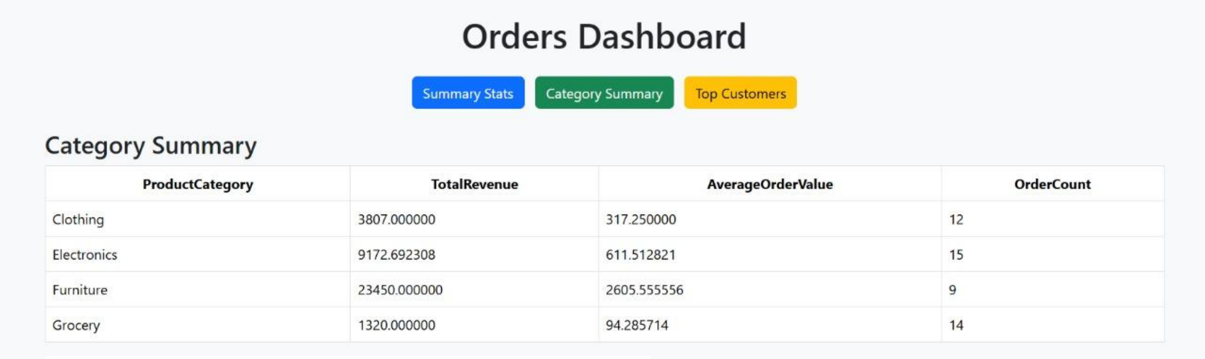


Orders Dashboard

Top 5 Customers

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





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