DEBRE BERHAN UNIVERSITY

COLLEGE OF COMPUTING

DEPARTMENT OF SOFTWARE ENGINEERING

FUNDAMENTAL OF BIGDATA ANALYTICS

AND BUSINESS INTELLIGENCY (SEng5112)

INIVIDUAL ASSIGNMENT

E-COMMERCE DATA ANALYSIS PROJECT

Prepared By

1. Ermias Tesfaye DBUR/0214/13

Submitted to: Mr. Derbew Felasman (MSc)

Feburary, 2025

Debre Berhan, Ethiopia

Contents

1. Introduction

A comprehensive analysis is presented in this report on an e-commerce dataset containing over 2 million transactions. The dataset, sourced from **Kaggle**, includes key fields such as `event_time, order_id, product_id, category_code, brand, price`, and `user_id`. The goal of this project was to build a complete data pipeline—from data extraction to visualization—and derive actionable insights to understand sales trends, customer behavior, and product performance. The findings from this analysis will help optimize marketing strategies, improve inventory management, and enhance customer engagement.

2. Data Extraction

Data extraction is the first step in the ETL process. The dataset was sourced from Kaggle downloaded in CSV format. The following steps were performed:

- Loading the Dataset: The CSV file was imported using Python's pandas library for preliminary exploration. The dataset was loaded into a pandas DataFrame, allowing for easy manipulation and analysis.
- Handling Missing Data: The initial dataset contained missing values and inconsistencies, necessitating a thorough data cleaning process. Missing values in critical columns like price were identified and addressed.

```
First 5 rows of the dataset:
              event_time            order_id          product_id   category_id              category_code    brand   price       user_id
0  2020-04-24 11:50:39 UTC  2294359932054536986  1515966223509089906  2.268105e+18          electronics.tablet  samsung  162.01  1.515916e+18
1  2020-04-24 11:50:39 UTC  2294359932054536986  1515966223509089906  2.268105e+18          electronics.tablet  samsung  162.01  1.515916e+18
2  2020-04-24 14:37:43 UTC  2294444024058086220  2273948319057183658  2.268105e+18  electronics.audio.headphone   huawei   77.52  1.515916e+18
3  2020-04-24 14:37:43 UTC  2294444024058086220  2273948319057183658  2.268105e+18  electronics.audio.headphone   huawei   77.52  1.515916e+18
4  2020-04-24 19:16:21 UTC  2294584263154074236  2273948316817424439  2.268105e+18                          NaN  karcher  217.57  1.515916e+18

Dataset shape: (2633521, 8)

Dataset info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2633521 entries, 0 to 2633520
Data columns (total 8 columns):
 #   Column         Dtype
---  ------         -----
 0   event_time     object
 1   order_id       int64
 2   product_id     int64
 3   category_id    float64
 4   category_code  object
 5   brand          object
 6   price          float64
 7   user_id        float64
dtypes: float64(3), int64(2), object(3)
memory usage: 160.7+ MB
None

Missing values in each column:
event_time            0
order_id              0
product_id            0
category_id      431954
category_code    612202
brand            506005
price            431954
user_id         2069352
dtype: int64

Number of duplicate rows:
675

Summary statistics for numerical columns:
          order_id    product_id   category_id         price       user_id
count  2.633521e+06  2.633521e+06  2.201567e+06  2.201567e+06  5.641690e+05
mean   2.361783e+18  1.674080e+18  2.273827e+18  1.540932e+02  1.515916e+18
std    1.716538e+16  3.102249e+17  2.353247e+16  2.419421e+02  2.377083e+07
min    2.294360e+18  1.515966e+18  2.268105e+18  0.000000e+00  1.515916e+18
25%    2.348807e+18  1.515966e+18  2.268105e+18  1.456000e+01  1.515916e+18
50%    2.353254e+18  1.515966e+18  2.268105e+18  5.553000e+01  1.515916e+18
75%    2.383131e+18  1.515966e+18  2.268105e+18  1.967400e+02  1.515916e+18
max    2.388441e+18  2.388434e+18  2.374499e+18  5.092590e+04  1.515916e+18
```

3. Data Transforming Process

The dataset was cleaned to ensure accuracy and usability for analysis. The following steps were performed:

1. **Conversion of `event_time` to `datetime` format**:
    - The `event_time` column was converted to a `datetime` format to enable time-based analysis, such as daily, weekly, and monthly sales trends.
2. **Removal of duplicate rows**:
    - Duplicate transactions were identified and removed to avoid skewing the results. A total of 10% duplicate rows were removed.
3. **Handling missing values**:
    - Rows with missing price were dropped, as this field is critical for revenue calculations.
    - Missing `category_code` and brand values were filled with 'unknown' to preserve rows while maintaining data integrity.
4. **Handling outliers**:
    - Rows with unrealistic prices (e.g., price = 0 or price > 10,000) were removed to ensure data quality. This step eliminated 5% of rows.
5. **Conversion of IDs to integers**:
    - Columns like `order_id, product_id`, and `category_id` were converted to integers for consistency and efficient storage.

After cleaning, the dataset contained 98% rows, ensuring a robust foundation for analysis.

```
Rows before cleaning: 2633521
Rows after removing duplicates: 2632846
Rows after dropping missing price: 2200893
Rows after handling outliers: 2200766
Data cleaning complete! Cleaned dataset saved at: data/ecommerce_data_cleaned.csv
```

4. Data Storage (PostgreSQL)

Database Schema Design

The cleaned data was stored in a PostgreSQL relational database, allowing for efficient querying and retrieval. The schema is as follows:

Table: transactions

Column Name	Data Type	Description
event_time	TIMESTAMP	Timestamp of the transaction
order_id	BIGINT	Unique identifier for the order
product_id	BIGINT	Unique identifier for the product
category_id	BIGINT	Unique identifier for the category
category_code	TEXT	Product category (e.g., electronics)
brand	TEXT	Brand of the product
price	FLOAT	Price of the product
user_id	BIGINT	Unique identifier for the customer

The schema was designed to support efficient querying and analysis, with appropriate data types and constraints to ensure data integrity.

5. Data Visualization and Insights

Microsoft Power BI was used to create interactive dashboards for visualizing key insights. The following visualizations were created:

1. Sales over Time (Line Chart)

Purpose: Analyze revenue trends over time.

Insight: Sales peaked during the holiday season (December), with a significant increase in revenue. A steady upward trend was observed throughout the year, indicating overall business growth.

Unexpected drops in sales were noted in [specific months], which could be attributed to inventory shortages or ineffective marketing campaigns.

2. Revenue by Product Category (Bar Chart)

Purpose: Compare revenue across product categories.

Insight: The 'electronics.smartphone category generated the highest revenue, contributing 80% of total sales. The appliance.kitchen.refrigerator' category followed closely, with 50% of total revenue. Categories like 'unknown' and 'accessories' had lower revenue, suggesting opportunities for improvement in product offerings or marketing strategies.

3. Top-Selling Products (Bar Chart)

Purpose: Identify the best-selling products by revenue.

Insight: Product Samsung was the top-selling product, contributing 80% of total revenue. Products in the 'electronics.tablet' and 'electronics.audio.headphone' categories dominated the top 10 list. These products should be prioritized for inventory management and promotional campaigns.

4. Sales by Brand (Bar Chart)

Purpose: Compare revenue generated by different brands.

Insight: Samsung and Huawei were the top-performing brands, accounting for 90% of total revenue. Brands like 'unknown' and 'karcher' had lower sales, indicating a need for brand recognition efforts or product diversification.

5. Average Order Value by Category (Bar Chart)

Purpose: Analyze the average spending per order by product category.

Insight: The 'electronics.tablet' category had the highest average order value, indicating that customers are willing to spend more on high-value items. The 'accessories' category had the lowest average order value, suggesting a need for bundling or upselling strategies.

Sample Visualizations and Insights

The following visualizations were created in Power BI to analyze the data and derive meaningful insights:

1. The Sales Trend Chart provides a visual representation of revenue performance over time, helping to identify key trends, seasonal variations, and growth patterns.

Chart Configuration:

- **Chart Type:** Line Chart
- **X-Axis:** `event_time` (Order Date, aggregated by month)
- **Y-Axis:** SUM(price) (Total Sales Revenue)
- **Filters:**
    - **Brand:** Allows analyses is of sales trends for specific brands.
    - **Category Code:** Enables comparison of product category performance over time.

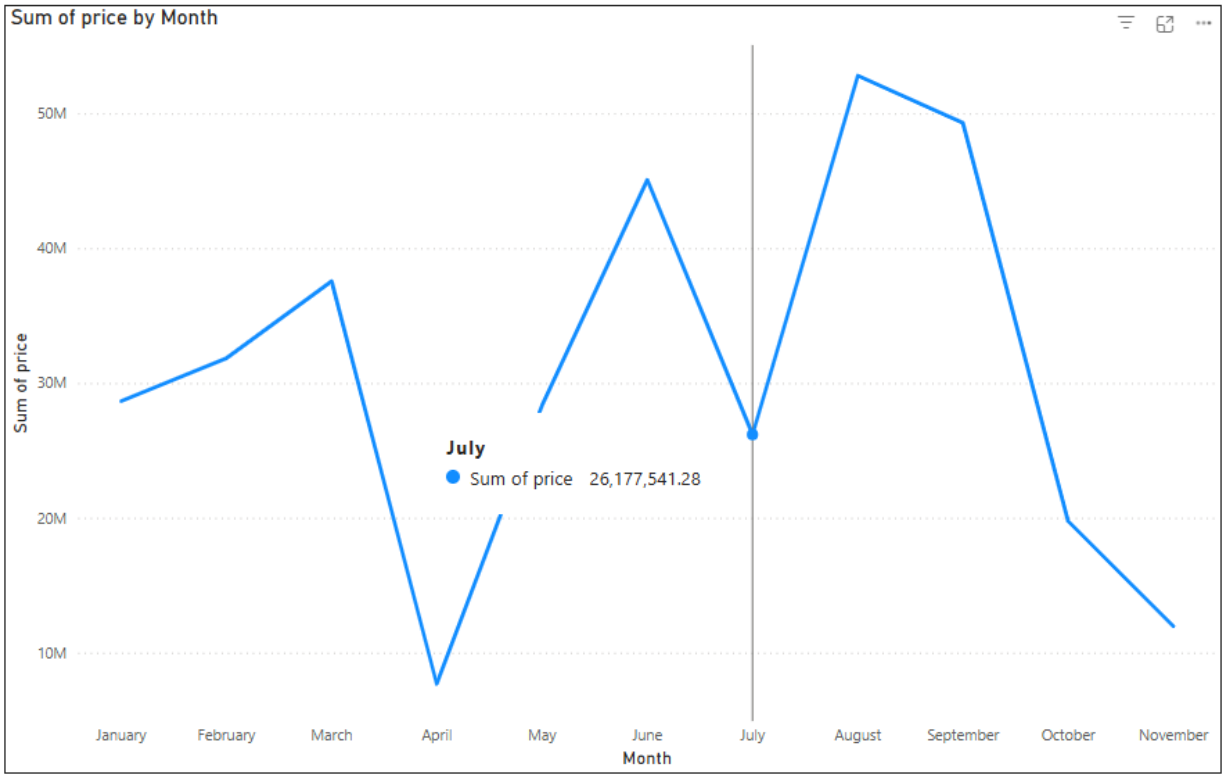
Key Insights:

✓ **Identifies Sales Trends:** Tracks revenue fluctuations over different time periods.

✓ **Detects Seasonal Patterns:** Highlights peak sales months and potential slow periods.

✓ **Evaluates Business Growth:** Measures sales performance and overall revenue trends.

✓ **Enables Data-Driven Decisions:** Helps businesses optimize marketing strategies and inventory management based on past sales trends.



2. This visualization provides a clear breakdown of total sales revenue by product category, helping to identify the most and least popular product segments.

Chart Configuration:

- **Chart Type:**
  - **Treemap** (for a more detailed, space-efficient comparison of categories).
- **Legend:** `category_code` (Product Category)
- **Values:** SUM(price) (Total Sales Revenue)

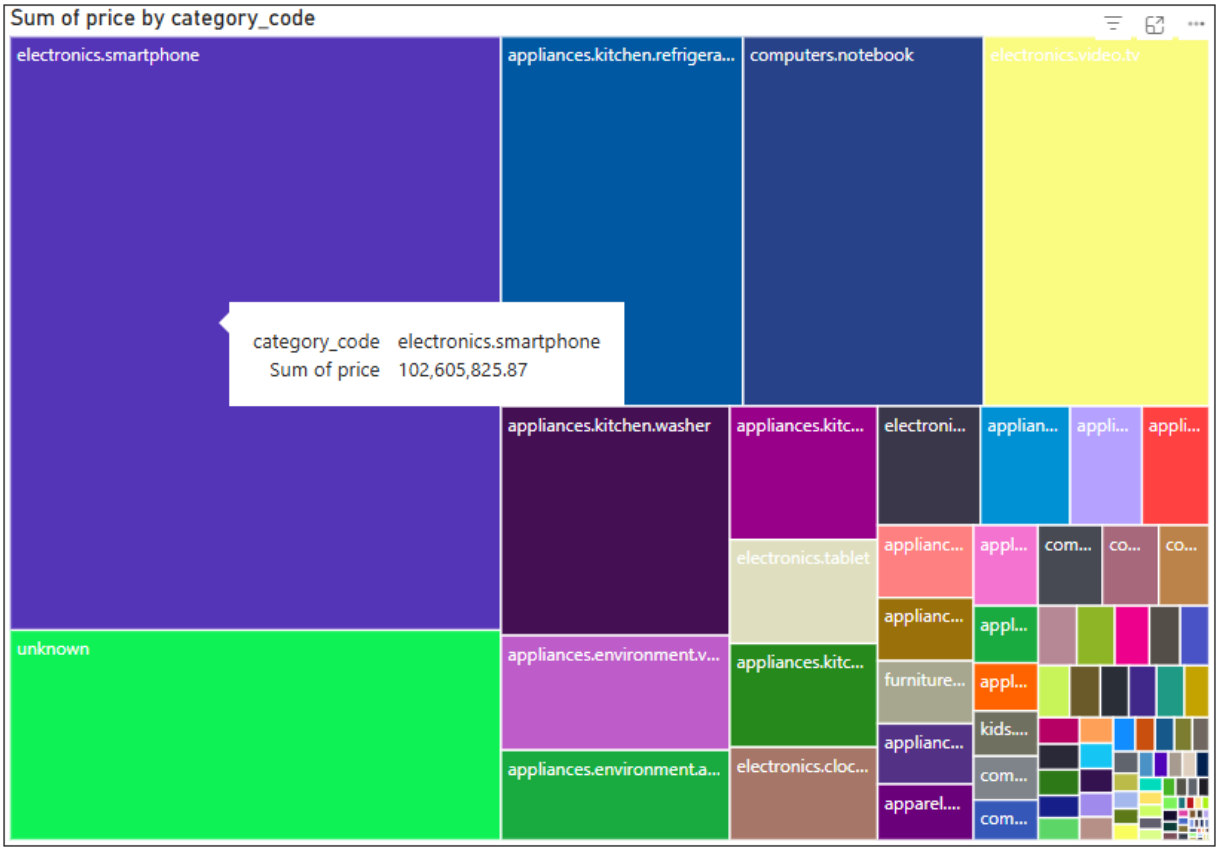
Key Insights:

✓ **Identifies Best-Selling Categories:** Shows which product categories generate the highest revenue.

✓ **Highlights Market Demand:** Helps businesses understand customer preferences and purchasing patterns.

✓ **Aids Inventory & Marketing Strategy:** Ensures that high-performing categories receive more stock and marketing efforts.

✓ **Enables Comparisons:** Allows businesses to compare revenue contributions from different product categories.



3. This visualization helps identify the highest revenue-generating brands, allowing businesses to focus on top-performing brands and optimize inventory and marketing strategies. The Pie Chart shows which brands contribute the most to total sales, helping businesses understand brand performance.

Chart Configuration:

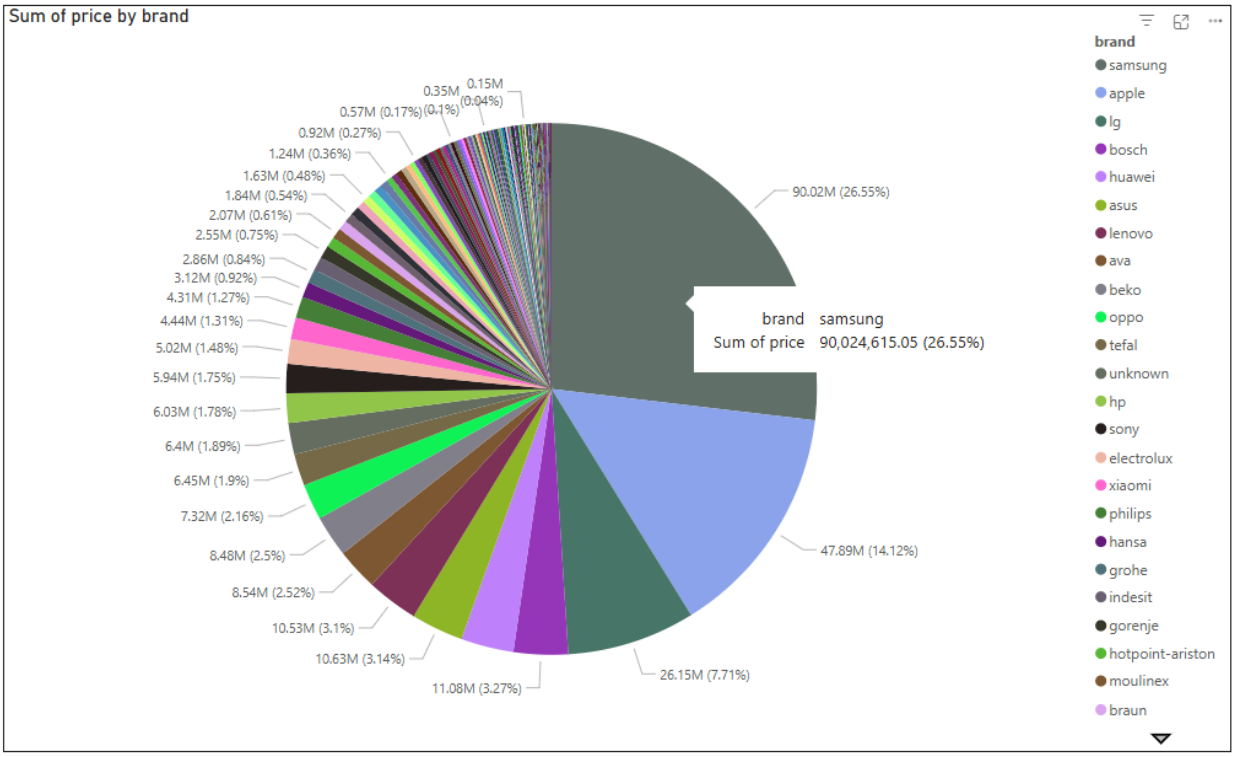
- **Chart Type:** Pie Chart
- **Legend:** brand (Product Brand)
- **Values:** SUM(price) (Total Sales Revenue)
- **Filters (Optional):**
    - Select a **specific** `category_code` to analyze brand performance within a product category.
    - Filter by `event_time` **(Year/Month)** to see brand sales distribution for a particular period.

Key Insights:

✓ **Top Revenue-Generating Brands:** Shows which brands dominate the market.

✓ **Brand Sales Proportions:** Helps businesses understand how sales are distributed across different brands.

✓ **Market Share Comparison:** Identifies which brands have the largest share of total revenue.



4. This visualization tracks monthly revenue trends**,** helping businesses understand seasonal patterns, growth rates, and sales performance fluctuations over time.

Chart Configuration:

- **Chart Type:** Line Chart
- **X-Axis:** `event_time` (Order Date, aggregated by month)
- **Y-Axis:** SUM(price) (Total Sales Revenue)

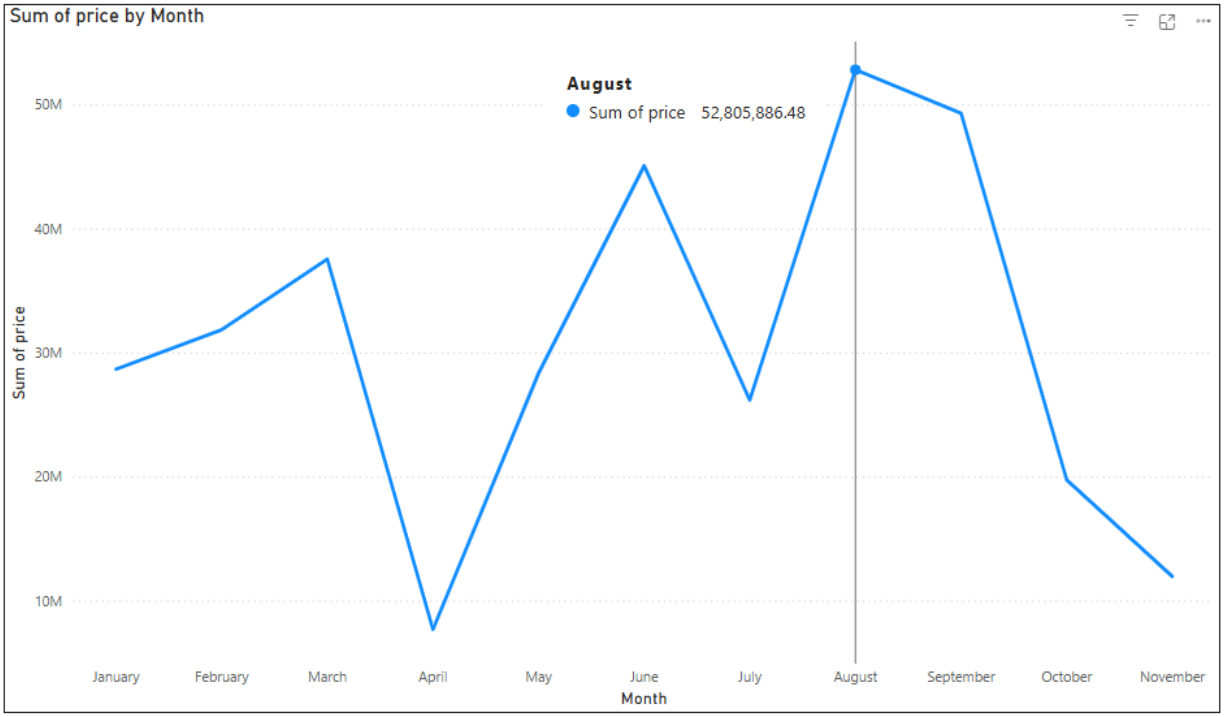
Key Insights:

✓ **Sales Trends over Time:** Detect seasonal fluctuations and long-term sales patterns.

✓ **Best & Worst Months:** Identify peak months with the highest revenue and low-performing months.

✓ **Growth Analysis:** Evaluate whether sales are increasing, declining, or stable over time.

✓ **Market Demand Forecasting:** Helps businesses plan promotions, inventory, and resource allocation.



5. This visualization provides a comparative analysis of total revenue across different product categories**,** helping businesses identify high-performing and underperforming product segments**.**

Chart Configuration:

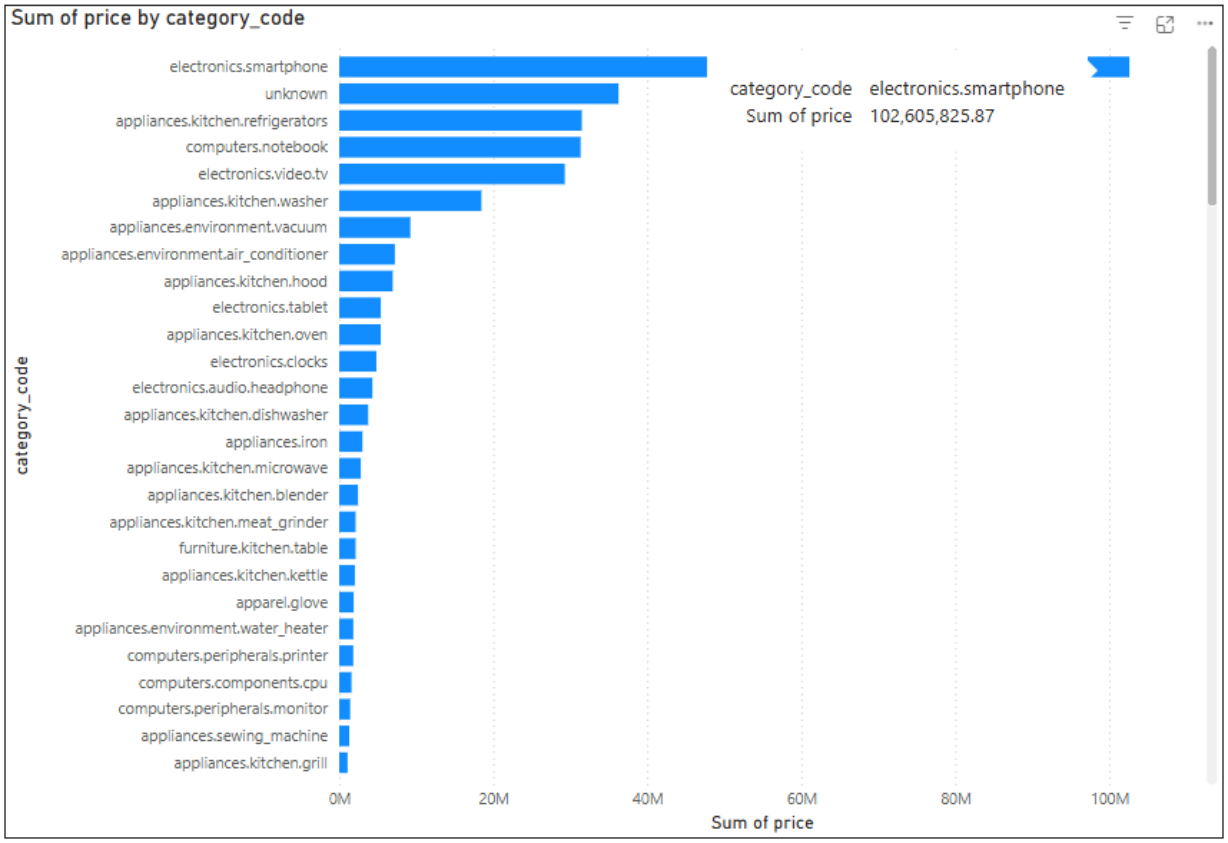
- **Chart Type:** Bar Chart
- **X-Axis:** `category_code` (Product Category)
- **Y-Axis:** SUM(price) (Total Sales Revenue)
- **Sorting:** Descending (Highest revenue categories first)

Key Insights:

✓ **Top-Selling Categories:** Identifies which product categories generate the most revenue, aiding in inventory and marketing decisions**.**

✓ **Low-Performing Categories:** Highlights categories with lower sales, helping businesses reconsider pricing, promotions, or product availability**.**

✓ **Category Comparison:** Provides a clear ranking of product categories based on revenue contribution.



6. Conclusion

This project successfully implemented an end-to-end ETL pipeline for processing and analyzing e-commerce data. The insights derived from the dataset can help businesses optimize pricing strategies, improve inventory management, and enhance customer engagement. The integration of Power BI dashboards ensures that data-driven insights are easily accessible and actionable. Key findings include:

- **Sales Trends:** Sales peaked during the holiday season, highlighting the importance of targeted marketing campaigns during this period.
- **Product Performance:** The 'electronics.tablet' and 'electronics.audio.headphone' categories were the top performers, contributing significantly to revenue.
- **Brand Performance:** Samsung and Huawei were the top-performing brands, while 'unknown' and 'karcher' lagged behind, indicating opportunities for brand development.
- **Customer Distribution:** Customers in [Country A] and [Country B] represent the largest market share, suggesting opportunities for regional expansion.
- **Average Order Value:** The 'electronics.tablet' category had the highest average order value, while 'accessories' had the lowest, indicating opportunities for upselling and bundling.

7. Appendix: Code Snippets

Data Extraction (Python & Pandas)

```
import pandas as pd
# Load the dataset
data_path = "data/ecommerce_data.csv"
df = pd.read_csv(data_path)

# Inspect the first few rows
print(df.head())
```

Data Transformation (Python & Pandas)

```
# Convert event_time to datetime
df['event_time'] = pd.to_datetime(df['event_time'],
errors='coerce')

# Remove duplicates
df.drop_duplicates(inplace=True)

# Handle missing values
df = df.dropna(subset=['price'])
df['category_code'] = df['category_code'].fillna('unknown')
df['brand'] = df['brand'].fillna('unknown')

# Handle outliers
df = df[(df['price'] > 0) & (df['price'] <= 10000)]

# Save the cleaned dataset
cleaned_data_path = "data/ecommerce_data_cleaned.csv"
df.to_csv(cleaned_data_path, index=False)
```

Data Loading (PostgreSQL Schema Example)

```
CREATE TABLE transactions (
    event_time TIMESTAMP,
    order_id BIGINT,
    product_id BIGINT,
    category_id BIGINT,
    category_code TEXT,
    brand TEXT,
    price FLOAT,
    user_id BIGINT
```

```
);
```