

E-COMMERCE INSIGHTS

In this analysis, we explored a dataset containing e-commerce transaction data using Python for data cleaning and SQL for analysis. The goal was to uncover key insights about customer demographics, product demand, sales trends, purchase behavior, and seller performance. Below are the findings and insights from each query.

DATA LOADING & EXPLORATION

1. Loading the Data

INPUT:

```
import pandas as pd
```

```
data = {
    'CustomerID': [12345, 12346, 12347, 12348, 12349, 12350, 12351, 12352, 12353, 12354],
    'OrderID': [1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010],
    'Product': ['T-shirt', 'Bicycle', 'Water bottle', 'Backpack', 'Sunglasses',
               'Notebook', 'Tablet', 'Smartphone', 'Camera', 'Headphones'],
    'Quantity': [2, 1, 5, 2, 1, 4, 1, 1, 2],
    'UnitPrice': [15.00, 200.00, 8.00, 45.00, 25.00, 12.00, 350.00, 700.00, 250.00, 75.00],
    'PurchaseDate': [
        '2022-07-15 08:30:00', '2022-07-15 09:00:00', '2022-07-16 10:00:00',
        '2022-07-16 11:30:00', '2022-07-17 12:45:00', '2022-07-18 13:00:00',
        '2022-07-19 14:25:00', '2022-07-20 15:45:00', '2022-07-21 16:00:00',
        '2022-07-22 17:35:00'
    ],
    'Country': ['USA', 'Canada', 'UK', 'USA', 'Australia', 'India',
               'Germany', 'France', 'Spain', 'Italy']
}
data
```

OUTPUT:

| | CustomerID | OrderID | Product | Quantity | UnitPrice | PurchaseDate | Country |
|---|------------|---------|--------------|----------|-----------|---------------------|-----------|
| 0 | 12345 | 1001 | T-shirt | 2 | 15.0 | 2022-07-15 08:30:00 | USA |
| 1 | 12346 | 1002 | Bicycle | 1 | 200.0 | 2022-07-15 09:00:00 | Canada |
| 2 | 12347 | 1003 | Water bottle | 5 | 8.0 | 2022-07-16 10:00:00 | UK |
| 3 | 12348 | 1004 | Backpack | 2 | 45.0 | 2022-07-16 11:30:00 | USA |
| 4 | 12349 | 1005 | Sunglasses | 1 | 25.0 | 2022-07-17 12:45:00 | Australia |
| 5 | 12350 | 1006 | Notebook | 4 | 12.0 | 2022-07-18 13:00:00 | India |
| 6 | 12351 | 1007 | Tablet | 1 | 350.0 | 2022-07-19 14:25:00 | Germany |
| 7 | 12352 | 1008 | Smartphone | 1 | 700.0 | 2022-07-20 15:45:00 | France |
| 8 | 12353 | 1009 | Camera | 1 | 250.0 | 2022-07-21 16:00:00 | Spain |
| 9 | 12354 | 1010 | Headphones | 2 | 75.0 | 2022-07-22 17:35:00 | Italy |

What We Did:

- We loaded the dataset into a Pandas DataFrame to begin the analysis.
- The dataset contains 7 columns, including CustomerID, OrderID, Product, UnitPrice, PurchaseDate, Country.

2. Data Exploration

INPUT:

```
df.info()
```

INSIGHT:

- The PurchaseDate column is stored as an object (string format) instead of a datetime type.
- This can lead to errors when performing date-based operations, such as filtering, sorting, or calculating time differences.

OUTPUT:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   CustomerID  10 non-null    int64  
 1   OrderID     10 non-null    int64  
 2   Product     10 non-null    object  
 3   Quantity    10 non-null    int64  
 4   UnitPrice   10 non-null    float64 
 5   PurchaseDate 10 non-null    object  
 6   Country     10 non-null    object  
dtypes: float64(1), int64(3), object(3)
memory usage: 692.0+ bytes
```

DATA CLEANING

1. Checking for Duplicates

INPUT:

```
duplicates = df.duplicated()
```

```
print('number of duplicates rows: ',  
duplicates.sum())
```

```
print(df[duplicates])
```

OUTPUT:

```
numbers of rows duplicated 0
```

WHY:

- There are no duplicate rows in the dataset.
- This ensures the data is clean and ready for analysis.

2. Updating PurchaseDate Datatype

INPUT:

```
df['PurchaseDate']=pd.to_datetime(df['  
PurchaseDate'])  
df['PurchaseDate']
```

```
dtype: datetime64[ns]
```

3. Saving data as a CSV file

INPUT:

```
ddf.to_csv('E-Commerce  
cleaned',index=False)
```

DATA ANALYSIS (SQL)

Filtering: 2.1)

1. Filter all transactions made by customers from a specific country (e.g., 'USA')

INPUT:

```
select * from transactions where
Country='USA';
```

OUTPUT:

| CustomerID | OrderID | Product | Quantity | UnitPrice | PurchaseDate | Country |
|------------|---------|----------|----------|-----------|---------------------|---------|
| 12345 | 1001 | T-shirt | 2 | 15.00 | 2022-07-15 08:30:00 | USA |
| 12348 | 1004 | Backpack | 2 | 45.00 | 2022-07-16 11:30:00 | USA |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

INISGHT:

- Analyzing the number of orders and total revenue from USA-based customers helps identify how much they contribute to overall sales.
- If sales are concentrated in specific regions, targeted marketing strategies can be implemented.

Recommendations:

- Expand to Similar Markets: If the USA shows strong sales, consider expanding to Canada or Mexico, which may have similar purchasing behavior.
- Monitor Customer Behavior: Track repeat purchases, peak buying seasons, and preferred payment methods to enhance customer experience.

2. Extract orders where the total spend (Quantity * UnitPrice) exceeds \$500

INPUT:

```
select * from transactions where
(Quantity * UnitPrice) > 500;
```

OUTPUT:

| CustomerID | OrderID | Product | Quantity | UnitPrice | PurchaseDate | Country |
|------------|---------|------------|----------|-----------|---------------------|---------|
| 12352 | 1008 | Smartphone | 1 | 700.00 | 2022-07-20 15:45:00 | France |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

INISGHT:

- These orders are likely placed by premium customers or bulk buyers.
- Certain products may frequently appear in these high-value transactions.

Recommendations:

- Offer exclusive discounts or VIP perks to customers with frequent high-value orders.
- If specific products are driving high-value sales, consider dynamic pricing to maximize profitability.

3. Identify customers who purchased more than 3 different products

INPUT:

```
select CustomerID , count(Product) as Productcount
from transactions
group by CustomerID
having Productcount >3;
```

OUTPUT:

| Result Grid | | Filter Rows: | |
|-------------|--------------|--------------|--|
| CustomerID | Productcount | | |

INISGHT:

- Customers who buy multiple products may be more receptive to personalized recommendations.
- Customer Loyalty & Retention if these customers frequently return to buy more, they can be targeted for loyalty programs.

Recommendations:

- Implement a customer rewards program to become high-value shoppers.
- Use targeted email or SMS campaigns to keep them engaged

2.2) 4. Filter transactions that occurred in July 2022

INPUT:

```
select CustomerID , count(Product) as Productcount
from transactions
group by CustomerID
having Productcount >3;
```

OUTPUT:

| Result Grid | | | | | | | Filter Rows: | Edit: | |
|-------------|---------|--------------|----------|-----------|---------------------|-----------|--------------|-------|--|
| CustomerID | OrderID | Product | Quantity | UnitPrice | PurchaseDate | Country | | | |
| 12345 | 1001 | T-shirt | 2 | 15.00 | 2022-07-15 08:30:00 | USA | | | |
| 12346 | 1002 | Bicycle | 1 | 200.00 | 2022-07-15 09:00:00 | Canada | | | |
| 12347 | 1003 | Water bottle | 5 | 8.00 | 2022-07-16 10:00:00 | UK | | | |
| 12348 | 1004 | Backpack | 2 | 45.00 | 2022-07-16 11:30:00 | USA | | | |
| 12349 | 1005 | Sunglasses | 1 | 25.00 | 2022-07-17 12:45:00 | Australia | | | |
| 12350 | 1006 | Notebook | 4 | 12.00 | 2022-07-18 13:00:00 | India | | | |
| 12351 | 1007 | Tablet | 1 | 350.00 | 2022-07-19 14:25:00 | Germany | | | |
| 12352 | 1008 | Smartphone | 1 | 700.00 | 2022-07-20 15:45:00 | France | | | |
| 12353 | 1009 | Camera | 1 | 250.00 | 2022-07-21 16:00:00 | Spain | | | |
| 12354 | 1010 | Headphones | 2 | 75.00 | 2022-07-22 17:35:00 | Italy | | | |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | | | |

INISGHT:

- The store attracts all kinds of shoppers, from those buying everyday items to those making big purchases.
- Budget-conscious customers go for things like T-shirts (\$30) and Notebooks (\$12), while others invest in high-end products like Smartphones (\$700) and Tablets (\$350).
- Sales peaked between July 15-20, likely because people shop more around payday.

Recommendations:

- Plan promotions or limited-time discounts around mid-month when shopping activity is highest to maximize sales.

5. Extract orders placed during weekends

INPUT:

```
select * from transactions where  
dayofweek(PurchaseDate) in (1,7);
```

INISGHT:

- This suggests that customers may prefer shopping on weekdays rather than weekends, or it could indicate a missed opportunity to boost weekend sales.
 - These products are related mainly to travel, outdoor, or leisure-related products, implying that customers might be shopping for weekend activities.

Recommendations:

- Launch Weekend-Only Promotions: Offer special deals on Fridays and Saturdays to encourage more weekend shopping,
 - for example: "Weekend Sale – 10% Off All Travel & Outdoor Gear!"

OUTPUT:

| Filter Rows: | | Edit: | | Export/Import: | | |
|--------------|---------|--------------|----------|----------------|---------------------|-----------|
| CustomerID | OrderID | Product | Quantity | UnitPrice | PurchaseDate | Country |
| 12347 | 1003 | Water bottle | 5 | 8.00 | 2022-07-16 10:00:00 | UK |
| 12348 | 1004 | Backpack | 2 | 45.00 | 2022-07-16 11:30:00 | USA |
| 12349 | 1005 | Sunglasses | 1 | 25.00 | 2022-07-17 12:45:00 | Australia |

6. Identify transactions during specific sales events, like Black Friday or Cyber Monday

INPUT:

```
select * from transactions  
where (month(PurchaseDate)=11 and  
dayofweek(PurchaseDate)=6 and  
day(PurchaseDate) between 23 and 29)  
or (month(PurchaseDate)=11 and  
dayofweek(PurchaseDate)=2 and  
day(PurchaseDate) between 26 and 30)
```

INISGHT:

- No Recorded Sales During Black Friday & Cyber Monday
 - The dataset does not contain any transactions on these critical sales dates.
 - This could indicate a lack of promotions, or a business model that does not capitalize on these events.

| Result Grid | | Filter Rows: | | Edit: | | Export/Import: | |
|-------------|---------|--------------|----------|-----------|--------------|----------------|--|
| CustomerID | OrderID | Product | Quantity | UnitPrice | PurchaseDate | Country | |
| HULL | NULL | NULL | NULL | NULL | NULL | NULL | |
| | | | | | | | |

Sorting:

1.Sort transactions

INPUT:

```
select Product,PurchaseDate ,(Quantity * UnitPrice) as totalspend
from transactions
order by totalspend desc,PurchaseDate asc,
Product asc;
```

INISGHT:

- The Smartphone (\$700) and Tablet (\$350) generated the highest revenue.
- Understanding these trends can help optimize stock availability and promotions.
- Products like Water Bottles (\$40) and Notebooks (\$48) are lower in value but likely contribute to steady sales.

Recommendations:

- Stock High-Demand, High-Value Products to Ensure products like Smartphones, Tablets, and Cameras are well-stocked.Improve early detection programs for high-risk patients.

2.Rank customers based on their total spending

INPUT:

```
select CustomerID, sum(Quantity * UnitPrice)
as totalspend,
Rank() over(order by sum(Quantity *
UnitPrice) desc) as spendingrank
from transactions
group by CustomerID
order by totalspend desc;
```

INISGHT:

- This analysis ranks customers based on their total spending, helping us identify top buyers and potential opportunities for targeted marketing.
- Top-Spending Customer Drive Sales is Customer 12352 is the highest spender with a total purchase value of \$700.

OUTPUT:

| Product | PurchaseDate | totalspend |
|--------------|---------------------|------------|
| Smartphone | 2022-07-20 15:45:00 | 700.00 |
| Tablet | 2022-07-19 14:25:00 | 350.00 |
| Camera | 2022-07-21 16:00:00 | 250.00 |
| Bicycle | 2022-07-15 09:00:00 | 200.00 |
| Headphones | 2022-07-22 17:35:00 | 150.00 |
| Backpack | 2022-07-16 11:30:00 | 90.00 |
| Notebook | 2022-07-18 13:00:00 | 48.00 |
| Water bottle | 2022-07-16 10:00:00 | 40.00 |
| T-shirt | 2022-07-15 08:30:00 | 30.00 |
| Sunglasses | 2022-07-17 12:45:00 | 25.00 |

OUTPUT:

| CustomerID | totalspend | spendingrank |
|------------|------------|--------------|
| 12352 | 700.00 | 1 |
| 12351 | 350.00 | 2 |
| 12353 | 250.00 | 3 |
| 12346 | 200.00 | 4 |
| 12354 | 150.00 | 5 |
| 12348 | 90.00 | 6 |
| 12350 | 48.00 | 7 |
| 12347 | 40.00 | 8 |
| 12345 | 30.00 | 9 |
| 12349 | 25.00 | 10 |

Aggregated Analysis:
1.Calculate the total amount spent by each customer and find the top 10 spenders
INPUT:

```
select CustomerID ,(Quantity * UnitPrice) as
totalspend from transactions
order by totalspend desc ,CustomerID
limit 10;
```

INISGHT:

- After the top three, spending gradually declines, with the lowest in the top 10 spending \$48.

Recommendations:

- Consider email campaigns or product recommendations based on past purchases to increase engagement.

OUTPUT:

| CustomerID | totalspend |
|------------|------------|
| 12352 | 700.00 |
| 12351 | 350.00 |
| 12353 | 250.00 |
| 12346 | 200.00 |
| 12354 | 150.00 |
| 12348 | 90.00 |
| 12350 | 48.00 |
| 12347 | 40.00 |
| 12345 | 30.00 |
| 12349 | 25.00 |

2.Group data by Country and analyze the total revenue generated per country
INPUT:

```
select Country ,(Quantity * UnitPrice) as
total_revenue from transactions
order by total_revenue desc ,Country;
```

INISGHT:

- France is a high-value market – Consider expanding marketing efforts and product availability here.
- UK & Australia have weak sales – It may be due to pricing, demand, or lack of awareness. A market study could uncover solutions.

Recommendations:

- Analyze customer behavior in different countries to tailor marketing strategies

OUTPUT:

| Country | total_revenue |
|-----------|---------------|
| France | 700.00 |
| Germany | 350.00 |
| Spain | 250.00 |
| Canada | 200.00 |
| Italy | 150.00 |
| USA | 120.00 |
| India | 48.00 |
| UK | 40.00 |
| Australia | 25.00 |

3.Identify which country has the highest average transaction value

INPUT:

```
select Country ,avg(Quantity * UnitPrice) as avg_transaction_value from transactions
group by Country
order by avg_transaction_value
limit 1;
```

INISGHT:

- Australia is leading the highest average transaction, with value equal to 25.

Recommendations:

- Implement a customer rewards program to retain high-value shoppers.

OUTPUT:

| Result Grid | | Filter Rows: |
|-------------|-----------|-----------------------|
| | Country | avg_transaction_value |
| ▶ | Australia | 25.000000 |

Product Analysis: 4.2)

1.Find the most purchased product and its total quantity sold

INPUT:

```
select Product, sum(quantity) as TQ_Sold
from transactions
group by Product
order by TQ_Sold desc
limit 1;
```

INISGHT:

- The Water Bottle is the most purchased product, with 5 units sold in total.

OUTPUT:

| Result Grid | | Filter Rows: |
|-------------|--------------|--------------|
| | Product | TQ_Sold |
| ▶ | Water bottle | 5 |

Recommendations:

- Diversify Inventory Promotion – Promote other products through cross-selling strategies to boost sales.

2. Identify the product that generated the highest revenue

INPUT:

```
select Product, sum(Quantity * UnitPrice) as
totalrevenue
from transactions
group by Product
order by totalrevenue desc
limit 1;
```

OUTPUT:

| Result Grid | | Filter Rows: |
|-------------|--------------|--------------|
| Product | totalrevenue | |
| Smartphone | 700.00 | |

INISGHT:

- Smartphones drive the most sales revenue, indicating strong customer demand.
- They may have higher profit margins or be frequently purchased in bulk.

Recommendations:

- Introduce bundle deals (e.g., phone + accessories) to encourage bigger purchases.
- Analyze customer behavior to understand why smartphones are leading and apply similar strategies to other products

3. Determine the top 3 least popular products based on sales quantity

INPUT:

```
select Product, sum(Quantity) as
totalsales_quantity
from transactions
group by Product
order by totalsales_quantity asc
limit 3;
```

OUTPUT:

| Result Grid | | Filter Rows: |
|-------------|---------------------|--------------|
| Product | totalsales_quantity | |
| Bicycle | 1 | |
| Sunglasses | 1 | |
| Tablet | 1 | |

INISGHT:

- Low demand or visibility – Customers might not be aware of these products or they may not meet their needs.
- Limited marketing efforts – If these products aren't promoted well, customers may not be interested.

Recommendations:

- Analyze customer feedback to understand if pricing, features, or availability are barriers to sales.
- Optimize website placement – Ensure these items appear in search results and recommendations

Time-Based Insights:
1. Analyze the total revenue generated per day in July 2022.
INPUT:

```
select date(PurchaseDate) as PurchaseDay,
       sum(quantity * UnitPrice) as totalrevenue
  from transactions
 where PurchaseDate between '2022-7-1' and '2022-7-31'
   group by PurchaseDay
  order by PurchaseDay asc, totalrevenue;
```

OUTPUT:

| PurchaseDay | totalrevenue |
|-------------|--------------|
| 2022-07-15 | 230.00 |
| 2022-07-16 | 130.00 |
| 2022-07-17 | 25.00 |
| 2022-07-18 | 48.00 |
| 2022-07-19 | 350.00 |
| 2022-07-20 | 700.00 |
| 2022-07-21 | 250.00 |
| 2022-07-22 | 150.00 |

INISGHT:

- The total revenue varies significantly across different days in July 2022.
- Best sales day: July 20, 2022 (\$700) – This suggests a strong demand for products on this day.
- Lowest sales day: July 17, 2022 (\$25) – This might indicate low customer activity.

Recommendations:

- Ensure adequate inventory and staffing on peak sales days to maximize revenue.
- Explore if external factors (e.g., weekends, holidays) affected sales.

2. Identify peak shopping hours based on the number of transactions.
INPUT:

```
select Hour(PurchaseDate) as Shopping_hours,
       count(OrderID) as transactions_count
  from transactions
 group by Shopping_hours
 order by transactions_count desc;
```

OUTPUT:

| Shopping_hours | transactions_count |
|----------------|--------------------|
| 8 | 1 |
| 9 | 1 |
| 10 | 1 |
| 11 | 1 |
| 12 | 1 |
| 13 | 1 |
| 14 | 1 |
| 15 | 1 |
| 16 | 1 |
| 17 | 1 |

INISGHT:

- The data shows that transactions are evenly spread across different hours of the day.
- There is no strong trend indicating a rush hour or peak period.

Recommendations:

- Running limited-time discounts or flash sales at specific hours can help identify which time frames attract the most shoppers.
- Running limited-time discounts or flash sales at specific hours can help identify which time frames attract the most shoppers.

3. Find the day with the highest total revenue.

INPUT:

```
select date(PurchaseDate) as PurchaseDay,
       sum(quantity * UnitPrice) as totalrevenue
  from transactions
 where PurchaseDate between '2022-7-1' and '2022-7-31'
 group by PurchaseDay
 order by totalrevenue desc
 limit 1;
```

OUTPUT:

| PurchaseDay | totalrevenue |
|-------------|--------------|
| 2022-07-20 | 700.00 |

INISGHT:

- Sales on this day were significantly higher than on other days.
- Analyzing what was sold, how many orders were placed, and the average order value can provide deeper insights.

Recommendations:

- If there was a promotion or special event, run a similar campaign on other days to boost revenue
- Use post-purchase follow-ups (emails, SMS) to keep customers engaged.

5.1. Create a column for total spend per transaction (Quantity * UnitPrice).

INPUT:

```
select *, (Quantity * UnitPrice) as TotalSpend
  from transactions;
```

OUTPUT:

| CustomerID | OrderID | Product | Quantity | UnitPrice | PurchaseDate | Country | TotalSpend | |
|------------|---------|--------------|----------|--------------|---------------------|---------------------|------------|-------|
| 12345 | 1001 | T-shirt | 2 | 15.00 | 2022-07-15 08:30:00 | USA | 30.00 | |
| 12346 | 1002 | Bicycle | 1 | 200.00 | 2022-07-15 09:00:00 | Canada | 200.00 | |
| 12347 | 1003 | Water bottle | 5 | 8.00 | 2022-07-16 10:00:00 | UK | 40.00 | |
| 12348 | 1004 | Backpack | 1 | Water bottle | 100 | 2022-07-16 11:30:00 | USA | 90.00 |
| 12349 | 1005 | Sunglasses | 1 | 25.00 | 2022-07-17 12:45:00 | Australia | 25.00 | |
| 12350 | 1006 | Notebook | 4 | 12.00 | 2022-07-18 13:00:00 | India | 48.00 | |
| 12351 | 1007 | Tablet | 1 | 350.00 | 2022-07-19 14:25:00 | Germany | 350.00 | |
| 12352 | 1008 | Smartphone | 1 | 700.00 | 2022-07-20 15:45:00 | France | 700.00 | |
| 12353 | 1009 | Camera | 1 | 250.00 | 2022-07-21 16:00:00 | Spain | 250.00 | |
| 12354 | 1010 | Headphones | 2 | 75.00 | 2022-07-22 17:35:00 | Italy | 150.00 | |

INISGHT:

- Some customers spent as little as \$25, while others made high-value purchases up to \$700.
- The highest-spending transaction was a smartphone purchase (\$700, France).

Recommendations:

- Promote premium items like smartphones, cameras, and tablets through discounts, installment plans, or bundles.
- Invest in localized promotions for high-spending regions (France, Germany, Canada)

5.2. Use quantiles to identify the top 10% of transactions based on total spend.

INPUT:

```
WITH TotalSpend AS (
    SELECT OrderID, CustomerID, Product,
           (Quantity * UnitPrice) AS TotalSpend,
           NTILE(10) OVER (ORDER BY (Quantity *
           UnitPrice) DESC) AS Bucket
      FROM Transactions
)
SELECT OrderID, CustomerID, Product,
       TotalSpend
  FROM TotalSpend
 WHERE Bucket = 1;
```

OUTPUT:

| OrderID | CustomerID | Product | TotalSpend |
|---------|------------|------------|------------|
| 1008 | 12352 | Smartphone | 700.00 |

INISHT:

- Unit Price & Total Revenue (+0.99): Higher prices lead to higher revenue.
- Quantity & Unit Price (-0.52): Expensive products sell in smaller quantities.
- Quantity & Total Revenue (-0.49): Selling more units doesn't always increase revenue.

Recommendations:

- Keep high-stock of Water bottles & Notebooks.
- Boost moderate sellers with discounts or promotions.

3. Analyze purchasing trends by country:

1. Identify the most popular product in each country.

INPUT:

```
select Product, count(OrderID) as
       counted_orders, Country
  from transactions
 group by Product, Country
 order by Country, counted_orders desc;
```

OUTPUT:

| Product | counted_orders | Country |
|--------------|----------------|-----------|
| Sunglasses | 1 | Australia |
| Bicycle | 1 | Canada |
| Smartphone | 1 | France |
| Tablet | 1 | Germany |
| Notebook | 1 | India |
| Headphones | 1 | Italy |
| Camera | 1 | Spain |
| Water bottle | 1 | UK |
| T-shirt | 1 | USA |
| Backpack | 1 | USA |

INISHT:

- Water bottles (25%) & Notebooks (20%) are the best sellers.
- T-shirts, Headphones, and Backpacks (10% each) sell moderately.
- Tablets, Sunglasses, Smartphones, Cameras, and Bicycles (5% each) sell the least.

Recommendations:

- Research: Check if data reflects a small sample or market trends.
- Promote: Use targeted campaigns to boost product popularity.

5.3.2 Determine the average order value per country.

INPUT:

```
select Product, AVG(Quantity * UnitPrice)
as AVG_ordervalue, Country
from transactions
group by Product, Country
order by AVG_ordervalue desc;
```

OUTPUT:

| Product | AVG_ordervalue | Country |
|--------------|----------------|-----------|
| Smartphone | 700.000000 | France |
| Tablet | 350.000000 | Germany |
| Camera | 250.000000 | Spain |
| Bicycle | 200.000000 | Canada |
| Headphones | 150.000000 | Italy |
| Backpack | 90.000000 | USA |
| Notebook | 48.000000 | India |
| Water bottle | 40.000000 | UK |
| T-shirt | 30.000000 | USA |
| Sunglasses | 25.000000 | Australia |

INISGHT:

- Countries with high-value purchases (France, Germany, Spain, Canada) could be targeted for premium product sales.
- Lower-value markets (USA, Australia, India, UK) may need more incentives to increase basket size.

Recommendations:

- Boost Premium Product Sales in High-Spending Countries; expand smartphone promotions in France, tablet deals in Germany, and high-end cameras in Spain.
- Leverage seasonal sales, influencer collaborations, and region-specific offers to boost engagement.

6. Visualization Python (using matplotlib) Heat map To Show correlation Between the Data

INPUT:

```
# Select numeric columns & calculate correlation
numeric_features =
df.select_dtypes(include=np.number).columns
corr_matrix = df[numeric_features].corr()

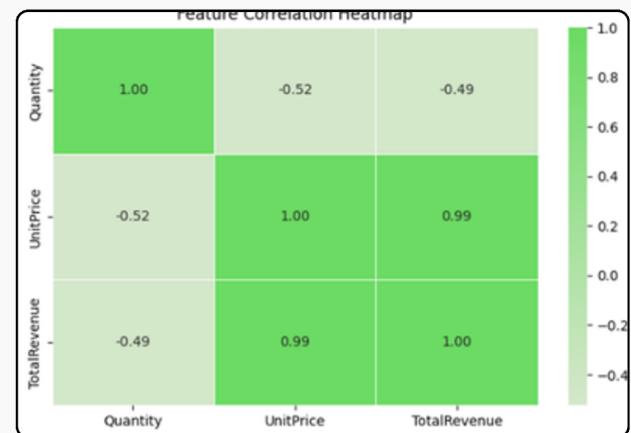
# Define custom colormap using the two colors
custom_cmap =
LinearSegmentedColormap.from_list("custom_cmap",
, ["#d5e7cb", "#6BDB63"])

# Plot heatmap
plt.figure(figsize=(8, 5))
sns.heatmap(corr_matrix, annot=True,
cmap=custom_cmap, fmt=".2f", linewidths=0.5)

# Title
plt.title("Feature Correlation Heatmap")

plt.show()
```

OUTPUT:



INISIGHT:

- Unit Price & Total Revenue (+0.99): Higher prices lead to higher revenue.
- Quantity & Unit Price (-0.52): Expensive products sell in smaller quantities.
- Quantity & Total Revenue (-0.49): Selling more units doesn't always increase revenue.

Recommendations:

- Adjust Pricing: Offer discounts on high-priced items to boost sales.
- Optimize Product Mix: Focus on premium products with high revenue potential.
- Sales Strategy: Use bundling or dynamic pricing to balance quantity and revenue

6. Visualization Python (using matplotlib)

1. Plot the total revenue per country using a bar chart.

INPUT:

```
# Group by country and sum the revenue
df['TotalRevenue'] = df['Quantity'] * df['UnitPrice']
revenue_per_country = df.groupby('Country')
['TotalRevenue'].sum().sort_values(ascending=False)

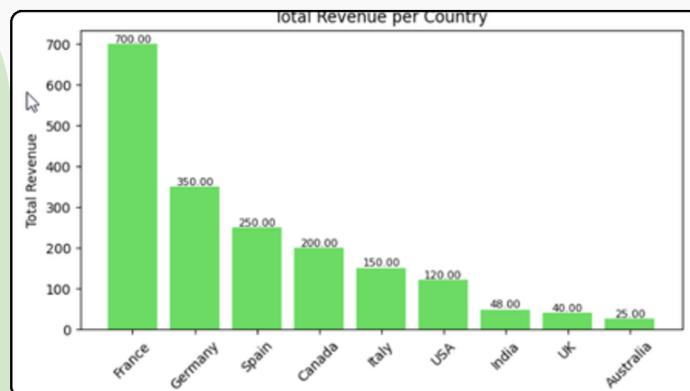
# Plot the sorted revenue per country
plt.figure(figsize=(8, 4))

# Assign the result of plt.bar to bars
bars = plt.bar(revenue_per_country.index,
revenue_per_country.values, color='darkred')
plt.xticks(rotation=90)

# Add data labels
for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval,
f'{yval:.2f}', ha='center', va='bottom', fontsize=8)

# Labels and title
plt.xlabel('Country')
plt.ylabel('Total Revenue')
plt.title('Total Revenue per Country')
plt.show()
```

OUTPUT:



INISGHT:

- Uneven Revenue: Some countries make much more money than others.
- Top Countries: One country makes 350.00 (highest), another 250.00.
- Low Performers: Some countries make very little, like 25.00.

Recommendations:

- Focus on Top Countries: Invest more in the highest earners to grow revenue.
- Improve Low Earners: Find out why some countries earn less and fix issues.
- Learn from Success: Copy strategies from top countries to help others grow.

6. Visualization Python (using matplotlib)

2. Visualize daily revenue trends with a line chart.

INPUT:

```
# plt size
plt.figure(figsize=(12, 8))

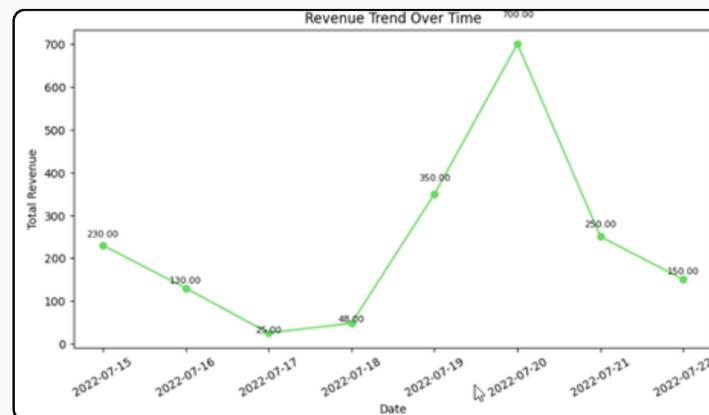
# group by revenue_trend
revenue_trend=df.groupby('Purchase_Date')
['TotalRevenue'].sum()

# visualize line chart
plt.plot(revenue_trend.index, revenue_trend.values,
marker='o', linestyle='-', color='b')

# Add data labels for revenue trend instead of
# revenue per country
for date, revenue in revenue_trend.items():
    plt.text(date, revenue + (revenue * 0.09),
f'{revenue:.2f}', ha='center', fontsize=8)

# labels
plt.xticks(rotation=30)
plt.xlabel('Date')
plt.ylabel('Total Revenue')
plt.title('Revenue Trend Over Time')
```

OUTPUT:



INISGHT:

- Fluctuating Revenue: Revenue varies significantly over time, with peaks and drops.
- Highest Point: The highest revenue was 350.00.
- Lowest Point: The lowest revenue was 48.00.

Recommendations:

- Analyze Peaks: Identify what caused high revenue points (e.g., promotions, events) and replicate successful strategies.
- Address Drops: Investigate reasons for low revenue periods and take corrective actions.
- Stabilize Revenue: Implement strategies to reduce fluctuations and maintain consistent revenue growth.

6. Visualization Python (using matplotlib)

3. Use a pie chart to show the distribution of products sold

INPUT:

```
# Group by both 'Country' and 'Product', summing
# the quantity sold

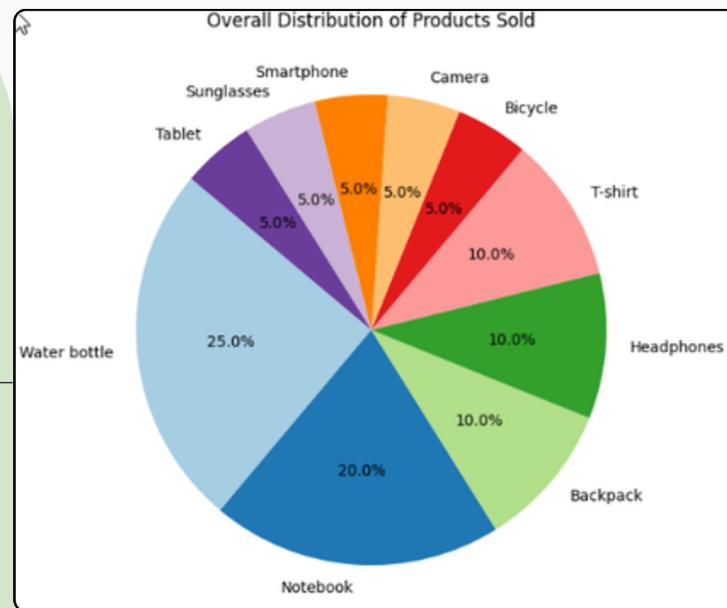
product_distribution = df.groupby('Product')
['Quantity'].sum()

# Aggregate across all countries
total_product_distribution =
product_distribution.groupby('Product').sum().sort_
values(ascending=False)

# Plot the pie chart
plt.figure(figsize=(7, 7))
plt.pie(total_product_distribution,
labels=total_product_distribution.index,
autopct='%.1f%%',
startangle=140, colors=plt.cm.Paired.colors)

# Title
plt.title('Overall Distribution of Products Sold')
plt.show()
```

OUTPUT:



INISGHT:

- Water bottles (25%) & Notebooks (20%) are the best sellers.
- T-shirts, Headphones, and Backpacks (10% each) sell moderately.
- Tablets, Sunglasses, Smartphones, Cameras, and Bicycles (5% each) sell the least.

Recommendations:

- Keep high-stock of Water bottles & Notebooks.
- Boost moderate sellers with discounts or promotions.
- Promote or bundle low-sellers to increase sales.
- Reevaluate low-selling products if demand stays low.