

1. Project Title

Customer Purchase Behavior Analysis Using Transactional Data (Pandas + Bootstrap Dashboard)

2. Project Description

This project analyzes customer purchase behavior of a retail store using transactional sales data. The goal is to clean and preprocess the dataset, identify patterns such as most purchased products, high-spending customers, purchase frequency, and revenue contribution by category.

The results are exported as CSV files and displayed in a professional dashboard built with Bootstrap that dynamically loads CSV data.

3. Team Member Details

NAME: K.Akshitha

USN:4GW23CI022

BRANCH: B.E CSE(AI&ML)

COLLEGE: GSSSIETW,Mysuru

Overview

The project performs analysis on a retail store's transactional dataset. The main objectives are:

1. Load and inspect dataset using Pandas
2. Clean the data (duplicates, missing values)
3. Compute $\text{TotalAmount} = \text{Quantity} * \text{Price}$ for each transaction
4. Aggregate data to find:
 - o Top 5 most purchased products
 - o Top spending customers
 - o Revenue contribution by product category
 - o Summary statistics including total transactions, total customers, average transaction value, and total revenue
5. Export all outputs to CSV files
6. Display results dynamically in a Bootstrap-based web dashboard

7. Use-Case Explanation

Use-Case ID Description

- UC1 Load and inspect transaction dataset
- UC2 Clean dataset (remove duplicates, handle missing values)
- UC3 Add TotalAmount column
- UC4 Identify top purchased products
- UC5 Identify high-spending customers
- UC6 Analyze revenue contribution by product categories
- UC7 Calculate summary statistics
- UC8 Display CSV outputs dynamically in a Bootstrap dashboard

6. Algorithms

1. Data Cleaning
 - Remove duplicates → `df.drop_duplicates()`
 - Handle missing values → `df.dropna()`
2. Calculating Total Amount
 - $\text{TotalAmount} = \text{Quantity} * \text{Price}$
3. Aggregations using Pandas
 - Top products →
`df.groupby("ProductID")["Quantity"].sum().sort_values(ascending=False).head(5)`
 - Top customers →
`df.groupby("CustomerID")["TotalAmount"].sum().sort_values(ascending=False)`
 - Revenue by product category →
`df.groupby("ProductCategory")["TotalAmount"].sum()`
4. Export CSVs
 - `df.to_csv("filename.csv", index=False)`

7. Code

```
# analysis.py

import pandas as pd

# Load dataset

df = pd.read_csv("transactions.csv")

# Clean column names

df.columns = df.columns.str.strip()

df.columns = [col.replace('\ufffe', '') for col in df.columns]

# Clean data

df.drop_duplicates(inplace=True)

df.dropna(inplace=True)

# Add TotalAmount column

df["TotalAmount"] = df["Quantity"] * df["Price"]

# Top 5 most purchased products

top_products = (

    df.groupby("ProductID")["Quantity"]

        .sum()

        .reset_index()

        .sort_values(by="Quantity", ascending=False)

        .head(5)

)

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

# Customers with highest spending

top_customers = (

    df.groupby("CustomerID")["TotalAmount"]

        .sum()

        .reset_index()

        .sort_values(by="TotalAmount", ascending=False)
```

```

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

# Revenue by Product Category
category_revenue = (
    df.groupby("ProductCategory")["TotalAmount"]
        .sum()
        .reset_index()
        .sort_values(by="TotalAmount", ascending=False)
)
category_revenue.to_csv("category_revenue.csv", index=False)

# Summary statistics
summary_stats = {
    "Total Transactions": [df["TransactionID"].nunique()],
    "Total Customers": [df["CustomerID"].nunique()],
    "Average Transaction Value": [round(df["TotalAmount"].mean(), 2)],
    "Total Revenue": [df["TotalAmount"].sum()]
}
pd.DataFrame(summary_stats).to_csv("summary_stats.csv", index=False)

print("✅ Analysis complete. All CSV files created.")

```

index.html (Bootstrap Dashboard, Tables Only)

```

<!DOCTYPE html>

<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Customer Purchase Behavior Dashboard</title>

```

```

<!-- Bootstrap CSS -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">

<!-- PapaParse for CSV reading -->
<script src="https://cdn.jsdelivr.net/npm/papaparse@5.4.1/papaparse.min.js"></script>

</head>

<body class="bg-light">

<div class="container my-5">
  <h1 class="text-center mb-5">📊 Customer Purchase Behavior Dashboard</h1>

  <!-- Tables Section -->
  <div id="tables"></div>
</div>

<script>
  // CSV files info
  const csvFiles = [
    {file: 'top_products.csv', title: 'Top 5 Most Purchased Products', theme: 'primary'},
    {file: 'top_customers.csv', title: 'Top Spending Customers', theme: 'success'},
    {file: 'category_revenue.csv', title: 'Revenue by Product Category', theme: 'warning'},
    {file: 'summary_stats.csv', title: 'Summary Statistics', theme: 'info'}
  ];

  const tablesDiv = document.getElementById('tables');

  // Function to render tables
  csvFiles.forEach(item => {

```

```
Papa.parse(item.file, {
  download: true,
  header: true,
  complete: function(results) {
    // Create table card
    const card = document.createElement('div');
    card.className = `card mb-4 shadow-sm`;

    const header = document.createElement('div');
    header.className = `card-header bg-${item.theme} text-white`;
    header.innerText = item.title;

    const body = document.createElement('div');
    body.className = 'card-body';

    const table = document.createElement('table');
    table.className = 'table table-striped table-bordered';

    const thead = document.createElement('thead');
    thead.className = 'table-dark';
    const headerRow = document.createElement('tr');
    Object.keys(results.data[0]).forEach(col => {
      const th = document.createElement('th');
      th.innerText = col;
      headerRow.appendChild(th);
    });
    thead.appendChild(headerRow);
    table.appendChild(thead);
  }
});
```

```
const tbody = document.createElement('tbody');

results.data.forEach(row => {
    const tr = document.createElement('tr');

    Object.values(row).forEach(val => {
        const td = document.createElement('td');
        td.innerText = val;
        tr.appendChild(td);
    });
    tbody.appendChild(tr);
});

table.appendChild(tbody);

body.appendChild(table);
card.appendChild(header);
card.appendChild(body);
tablesDiv.appendChild(card);

});

});

});

</script>

<!-- Bootstrap JS -->

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>
```

8. Explanation of the Code

1. **analysis.py** reads CSV, cleans data, calculates totals, aggregates metrics, and exports CSVs.
 2. **index.html** reads the CSVs dynamically using **PapaParse** and displays them in professional Bootstrap tables.
 3. Dashboard Layout:
 - o Each CSV file is displayed in a **card with a title**.
 - o Tables are striped and bordered for readability.
-

9. Dashboard Layout

- Cards for each CSV file:
 - o Top 5 Products
 - o Top Customers
 - o Revenue by Category
 - o Summary Statistics
 - Bootstrap ensures a clean, responsive, and professional look.
-

10. Output

The screenshot shows a development environment with multiple windows open:

- Terminal:** Shows the execution of `analysis.py`. The code reads a CSV file, strips column names, removes leading/trailing spaces from columns, prints the first 5 rows, prints descriptive statistics, and drops duplicates. The output shows the creation of four CSV files: `top_products.csv`, `top_customers.csv`, `category_revenue.csv`, and `summary_stats.csv`. The average transaction value is printed as 299.6.
- Copilot Workspace:** A sidebar titled "Welcome to Copilot" with options to "Ask a question", "Build workspace", and "Show project config". It also displays the output of the analysis script.
- File Explorer:** Shows the project structure with files like `analysis.py`, `transaction.csv`, and `index.html`.
- Status Bar:** Shows the current session details: In 17 Cell(s), Spans 2, Uptime 00:00:00, Python 3.10.7, and Port 5500.

Customer Purchase Behavior Dashboard

Top 5 Most Purchased Products

ProductID	Quantity
P001	77
P005	24
P002	3
P004	3
P003	3

Revenue by Product Category

ProductCategory	TotalAmount
Furniture	25600
Electronics	17150
Clothing	5390

127.0.0.1:5500/index.html

Electronics	17150
Clothing	5390
Grocery	1540

Top Spending Customers

CustomerID	TotalAmount
C131	3900
C114	3600
C139	2600
C110	2500
C127	2400
C135	2300
C119	2200
C102	2150
C123	2100
C143	2000
C105	2000

C128	180
C107	160
C117	150
C141	150
C120	140
C144	120
C124	100
C140	100
C132	80
C116	80
C113	60
C136	60

Summary Statistics			
Total Transactions	Total Customers	Average Transaction Value	Total Revenue
50	44	993.6	49680

11. Bibliography

This project demonstrates **Pandas-based analysis** and **Bootstrap table visualization**. It provides insights into customer behavior and revenue patterns, all without using charts. Exported CSVs are suitable for reporting or further analysis

1. Python Documentation – <https://docs.python.org/3/>
2. Pandas Documentation – <https://pandas.pydata.org/>
3. Bootstrap Documentation – <https://getbootstrap.com/>
4. PapaParse JS – <https://www.papaparse.com/>