

Customer Purchase Behavior Analysis Using Transactional Data

Pandas-Based

Project Title

Customer Purchase Behavior Analysis Using Transactional Data

Problem Statement

A retail store aims to understand customer purchasing patterns using its transactional sales data. The objective is to clean and analyze the dataset using Pandas to uncover insights such as top-selling products, high-spending customers, and purchase frequency—without relying on visualizations.

Objectives

1. Clean the dataset: handle missing values, duplicates, and ensure correct data types.
 2. Perform exploratory analysis using Pandas.
 3. Identify key metrics:
 - o Top 5 most purchased products
 - o Customers with the highest total spending
 - o Average purchase amount per transaction
 - o Purchase frequency per customer
 - o Product categories contributing most to revenue
 4. Export aggregated results to CSV for reporting.
-

Team Member

Priya N Singh

Full Stack Developer & Data Analyst

Skilled in Pandas, Flask, CSS, HTML, and modular Python scripting. Passionate about clarity, reproducibility, and elegant design in analytics workflows.

Index

1. Overview
 2. Dataset Description
 3. Data Cleaning
 4. Exploratory Analysis
 5. Use Case Explanations
 6. Algorithms
 7. UML Diagrams
 8. Front-End Design (Optional)
 9. Code & Explanation
 10. Screenshots & Output
 11. Closure & Bibliography
-

1. Overview

This project demonstrates how transactional data can be transformed into actionable insights using Pandas. It emphasizes modular, reproducible workflows and clean data handling, making it ideal for onboarding, teaching, and business reporting.

2. Dataset Description

Example File: transactions.csv

Columns:

- TransactionID
- CustomerID
- ProductID
- ProductCategory
- Quantity
- Price
- TransactionDate

Sample Entry:

T001, C101, P501, Electronics, 1, 500, 2023-01-05

3. Data Cleaning:

Performed using Pandas:

```
df = pd.read_csv('transactions.csv')  
df.drop_duplicates(inplace=True)  
df.dropna(inplace=True)  
df['TransactionDate'] = pd.to_datetime(df['TransactionDate'])  
df['TotalAmount'] = df['Quantity'] * df['Price']
```

4. Exploratory Analysis

- df.head(), df.info(), df.describe()
 - Added TotalAmount column
 - Verified data types and integrity
-

5. Use Case Explanations

Top 5 Most Purchased Products

```
top_products = df.groupby('ProductID')['Quantity'].sum().sort_values(ascending=False).head(5)
```

Top Spending Customers

```
top_customers=df.groupby('CustomerID')['TotalAmount'].sum().sort_values(ascending=False).head(5)
```

Average Transaction Value

```
avg_transaction = df['TotalAmount'].mean()
```

Purchase Frequency per Customer

```
purchase_freq = df['CustomerID'].value_counts()
```

Revenue by Product Category

```
category_revenue =  
df.groupby('ProductCategory')['TotalAmount'].sum().sort_values(ascending=False)
```

6. Algorithms

- **TotalAmount Calculation**
TotalAmount = Quantity * Price
- **GroupBy Aggregations**
Used groupby() with sum() and count()
- **Sorting & Exporting**
.sort_values() and .to_csv() for reporting

7. UML Diagrams

Use-Case Diagram

- Actor: Analyst
- Use Cases: Load Data, Clean Data, Analyze, Export

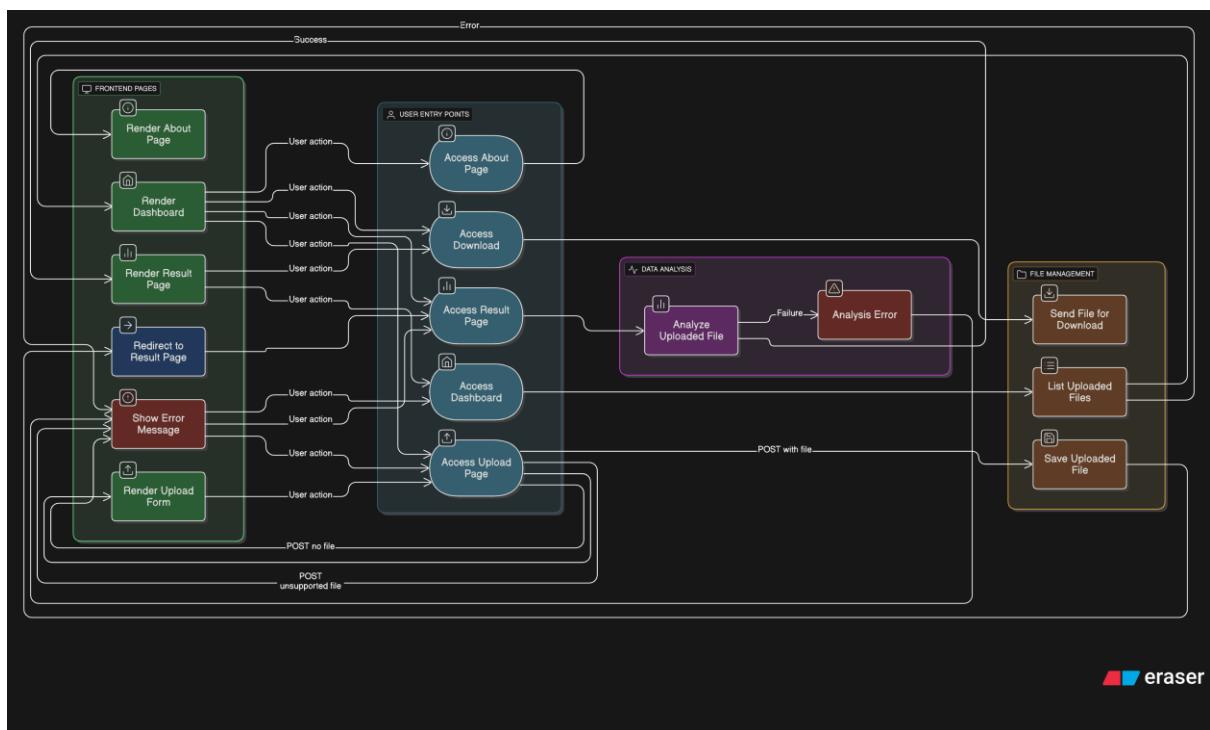
Activity Diagram

- Start → Load → Clean → Analyze → Export → End

Data-Flow Diagram

- Input: CSV → Processing: Pandas → Output: Aggregated CSVs

Working of `app.py` with UML diagram:



8. Front-End Design (HTML, FLASK & CSS)

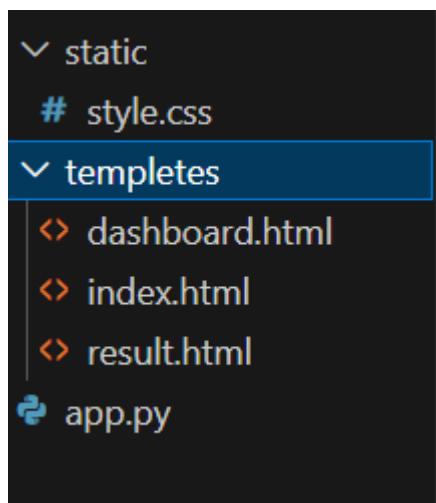
If integrated into a Flask app:

- Upload interface
 - Summary display
 - Download buttons for CSV exports
- Styled with Bootstrap for clarity and responsiveness
-

9. Code & Explanation

How to save the file in VS Code:

```
seals_analysis ( folder )
|
|---- static (folder)
|      └── style.css (file)      # Custom CSS for styling your web pages
|
|---- template (folder)        # (Typo: should be 'templates')
|      └── index.html          # Landing page for file upload or instructions
|      └── dashboard.html       # Possibly a summary or navigation page
|      └── result.html         # Displays processed data and insight
|
|----- app.py (file) # Main Flask application script
```



Code:

app.py

```
from flask import Flask, render_template, request, redirect, url_for, send_file, flash
import pandas as pd
import os

app = Flask(__name__)
app.secret_key = "your_secret_key"
UPLOAD_FOLDER = "uploads"
os.makedirs(UPLOAD_FOLDER, exist_ok=True)

def load_data(filepath):
    ext = os.path.splitext(filepath)[1].lower()
    if ext == ".csv":
        return pd.read_csv(filepath)
    elif ext in [".xls", ".xlsx"]:
        return pd.read_excel(filepath)
    elif ext == ".json":
        return pd.read_json(filepath)
    elif ext == ".xml":
        return pd.read_xml(filepath)
    else:
        raise ValueError("Unsupported file format")

def analyze_data(filepath):
    df = load_data(filepath)
    df.columns = [col.strip() for col in df.columns]
    df.drop_duplicates(inplace=True)

    try:
        df['TransactionDate'] = pd.to_datetime(df['TransactionDate'], errors='coerce')
        df['TotalAmount'] = df['Quantity'] * df['Price']
    except KeyError as e:
        raise ValueError(f"Missing expected column: {e}")


```

```

top_products = df.groupby('ProductID')['Quantity'].sum().sort_values(ascending=False).head(5)

top_customers = df.groupby('CustomerID')['TotalAmount'].sum().sort_values(ascending=False).head(5)

category_revenue = df.groupby('ProductCategory')['TotalAmount'].sum().sort_values(ascending=False)

summary_stats = {

    "Total Transactions": len(df),

    "Total Revenue": df['TotalAmount'].sum(),

    "Average Transaction Value": round(df['TotalAmount'].mean(), 2),

    "Unique Customers": df['CustomerID'].nunique(),

    "Unique Products": df['ProductID'].nunique()

}

return {

    "summary": pd.DataFrame(list(summary_stats.items()), columns=["Metric",
    "Value"]).to_html(classes="table table-bordered text-center align-middle", index=False),

    "top_products": top_products.reset_index().to_html(classes="table table-striped table-bordered text-
center align-middle", index=False),

    "top_customers": top_customers.reset_index().to_html(classes="table table-striped table-bordered text-
center align-middle", index=False),

    "category_revenue": category_revenue.reset_index().to_html(classes="table table-striped table-bordered
text-center align-middle", index=False)

}

@app.route("/")
@app.route("/dashboard")

def dashboard():

    try:

        files = os.listdir(UPLOAD_FOLDER)

        files = [f for f in files if f.lower().endswith((".csv", ".xls", ".xlsx", ".json", ".xml"))]

    except Exception as e:

        files = []

        flash(f"Error loading files: {e}")

    return render_template("dashboard.html", files=files)

```

```

@app.route("/upload", methods=["GET", "POST"])

def upload():
    if request.method == "POST":
        file = request.files.get("file")
        if not file:
            flash("No file selected.")
        elif not file.filename.lower().endswith((".csv", ".xls", ".xlsx", ".json", ".xml")):
            flash("Unsupported file format.")
        else:
            filepath = os.path.join(UPLOAD_FOLDER, file.filename)
            file.save(filepath)
            return redirect(url_for("result", filename=file.filename))
    return render_template("index.html")

@app.route("/result/<filename>")
def result(filename):
    filepath = os.path.join(UPLOAD_FOLDER, filename)
    try:
        data = analyze_data(filepath)
        return render_template("result.html", data=data, filename=filename)
    except Exception as e:
        return f"<h2>Error:</h2><pre>{e}</pre>"

@app.route("/download/<filename>")
def download(filename):
    filepath = os.path.join(UPLOAD_FOLDER, filename)
    return send_file(filepath, as_attachment=True)

@app.route("/about")
def about():
    return render_template("about.html")

```

```
if __name__ == "__main__":
    app.run(debug=True)
```

dashboard.html

```
<!DOCTYPE html>

<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Data Dashboard</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
<style>
    body {
        font-family: 'Inter', sans-serif;
        background: linear-gradient(135deg, #868993 0%, #0c463b 100%);
        min-height: 100vh;
        padding: 2rem 0;
        color: #495057;
    }
    .dashboard-card {
        background: rgba(255, 255, 255, 0.95);
        border-radius: 16px;
        box-shadow: 0 10px 30px rgba(0, 0, 0, 0.15);
        padding: 2.5rem;
        margin: 0 auto;
        max-width: 800px;
        backdrop-filter: blur(10px);
    }
}
```

```
}

h1 {
    color: #2d3748;
    font-weight: 700;
    margin-bottom: 1rem;
}

.header-icon {
    font-size: 2.5rem;
    margin-bottom: 1rem;
    color: #3e4460;
}

.subtitle {
    color: #6c757d;
    margin-bottom: 2rem;
}

.file-item {
    background: #f8f9fa;
    border-radius: 12px;
    border: none;
    margin-bottom: 1rem;
    padding: 1rem 1.5rem;
    transition: all 0.3s ease;
}

.file-item:hover {
    background: #e9ecf;
    transform: translateY(-2px);
    box-shadow: 0 4px 8px rgba(0,0,0,0.05);
}

.btn-view {
    background: linear-gradient(135deg, #464954 0%, #764ba2 100%);
    border: none;
    color: white;
    border-radius: 8px;
```

```

padding: 0.4rem 1rem;
}

.btn-download {
border-radius: 8px;
padding: 0.4rem 1rem;
}

.btn-upload {
background: linear-gradient(135deg, #28a745 0%, #20c997 100%);
border: none;
border-radius: 10px;
padding: 0.6rem 1.5rem;
font-weight: 600;
margin-top: 1.5rem;
}

.footer {
color: #6c757d;
margin-top: 2.5rem;
font-size: 0.9rem;
}


```

</style>

</head>

<body>

```

<div class="dashboard-card text-center">
<div class="header-icon">
<i class="fas fa-chart-line"></i>
</div>
<h1>Data Analysis Of Sales Dashboard</h1>
<p class="subtitle">Explore uploaded files, view insights, and download summaries</p>


```

{% with messages = get_flashed_messages() %}

{% if messages %}

```

<div class="alert alert-warning alert-dismissible fade show">
{{ messages[0] }}
```

```

<button type="button" class="btn-close" data-bs-dismiss="alert"></button>
</div>
{% endif %}
{% endwith %}

{% if files %}

<div class="list-group text-start my-4">
  {% for file in files %}
    <div class="file-item list-group-item d-flex justify-content-between align-items-center">
      <span><i class="fas fa-file-alt me-2"></i>{{ file }}</span>
      <div>
        <a href="{{ url_for('result', filename=file) }}" class="btn btn-sm btn-view me-2">
          <i class="fas fa-eye me-1"></i>View
        </a>
        <a href="{{ url_for('download', filename=file) }}" class="btn btn-sm btn-outline-secondary btn-download">
          <i class="fas fa-download me-1"></i>Download
        </a>
      </div>
    </div>
  {% endfor %}
</div>
{% else %}

<div class="my-5 py-3">
  <i class="fas fa-folder-open" style="font-size: 3rem; color: #dee2e6;"></i>
  <p class="text-muted mt-3">No files uploaded yet</p>
</div>
{% endif %}

<a href="{{ url_for('upload') }}" class="btn btn-upload">
  <i class="fas fa-cloud-upload-alt me-2"></i>Upload New File
</a>

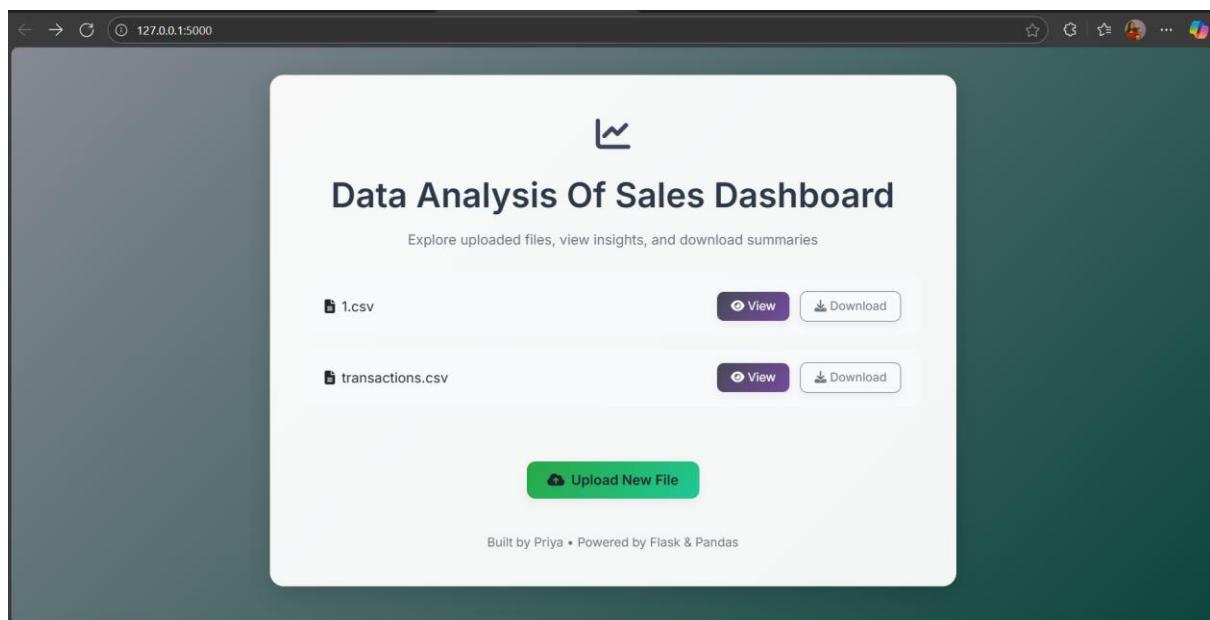
```

```

<div class="footer">
    Built by Priya • Powered by Flask & Pandas
</div>
</div>

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

```



index.html

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>Upload Your Transactions CSV</title>
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <!-- Bootstrap & Fonts -->
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
        <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600&display=swap" rel="stylesheet">
    </head>
    <body>
        <div class="header">
            <h1>Data Analysis Of Sales Dashboard</h1>
            <p>Explore uploaded files, view insights, and download summaries</p>
        </div>
        <div class="content">
            <table border="1">
                <thead>
                    <tr>
                        <th>File Name</th>
                        <th>Actions</th>
                    </tr>
                </thead>
                <tbody>
                    <tr>
                        <td>1.csv</td>
                        <td><button>View</button> <button>Download</button></td>
                    </tr>
                    <tr>
                        <td>transactions.csv</td>
                        <td><button>View</button> <button>Download</button></td>
                    </tr>
                </tbody>
            </table>
            <div>
                <button>Upload New File</button>
            </div>
        </div>
        <div class="footer">
            Built by Priya • Powered by Flask & Pandas
        </div>
    </body>
</html>

```

```
<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}>
```

```
<style>
```

```
body {  
    font-family: 'Inter', sans-serif;  
    background-color: #f4f6f8;  
    color: #343a40;  
}
```

```
.upload-card {  
    max-width: 500px;  
    margin: 5rem auto;  
    background: white;  
    padding: 2.5rem;  
    border-radius: 16px;  
    box-shadow: 0 8px 24px rgba(0, 0, 0, 0.05);  
}
```

```
h1 {
```

```
    font-weight: 600;  
    font-size: 1.8rem;  
}
```

```
.form-label {
```

```
    font-weight: 500;  
}
```

```
.form-control {
```

```
    border-radius: 8px;  
    padding: 0.6rem;  
}
```

```
.btn-primary {
```

```

border-radius: 8px;
font-weight: 600;
padding: 0.6rem;
background: linear-gradient(135deg, #0d6efd, #3d8bfd);
border: none;
transition: all 0.3s ease;

}

.btn-primary:hover {
background: linear-gradient(135deg, #3d8bfd, #0d6efd);
box-shadow: 0 6px 14px rgba(13, 110, 253, 0.3);
transform: translateY(-2px);
}

.alert {
border-radius: 8px;
font-size: 0.95rem;
}

.footer {
font-size: 0.85rem;
color: #6c757d;
margin-top: 2rem;
}

</style>
</head>
<body>
<div class="upload-card text-center">
<h1 class="mb-3">📁 Upload Your Transactions CSV</h1>
<p class="text-muted mb-4">Select your CSV file to analyze your transactions instantly.</p>
<br/>
<% with messages = get_flashed_messages() %>
<% if messages %>

```

```

<div class="alert alert-warning shadow-sm">{{ messages[0] }}</div>

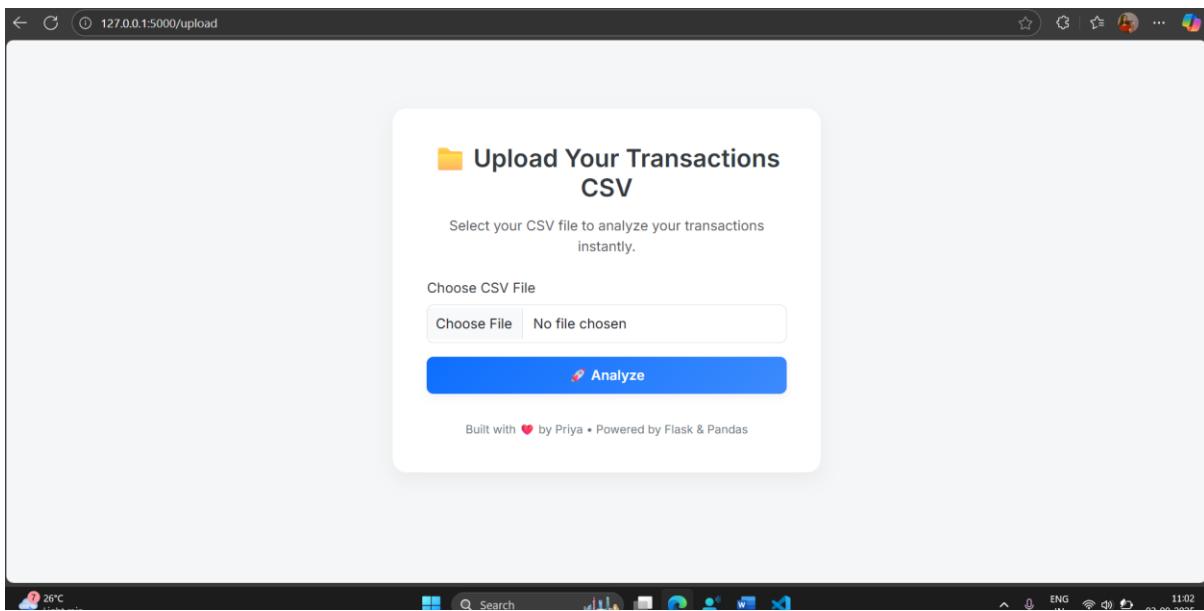
{% endif %}

{% endwith %}

<form method="POST" enctype="multipart/form-data" action="{{ url_for('upload') }}">
  <div class="mb-3 text-start">
    <label for="file" class="form-label">Choose CSV File</label>
    <input type="file" id="file" class="form-control" name="file" accept=".csv" required>
  </div>
  <button type="submit" class="btn btn-primary w-100">🚀 Analyze</button>
</form>

<div class="footer">
  Built with ❤️ by Priya • Powered by Flask & Pandas
</div>
</div>
</body>
</html>

```



Result.html

```
<!DOCTYPE html>

<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Analysis Results</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
<style>
    body{font-family:'Inter',sans-serif;background:linear-gradient(135deg,#f5f7fa 0%,#c3cfe2 100%);min-height:100vh;padding-top:2rem;color:#2d3748;}

    .container{max-width:900px;background:white;border-radius:18px;box-shadow:0 10px 30px rgba(0,0,0,0.08);padding:2rem;margin-bottom:2rem;}

    h2{font-weight:700;margin-bottom:1.5rem;color:#2d3748;padding-bottom:0.5rem;border-bottom:2px solid rgba(66,153,225,0.3);}

    .filename{color:#4299e1;background:rgba(66,153,225,0.1);padding:0.5rem 1rem;border-radius:8px;font-weight:600;}

    .section-btn{width:100%;text-align:left;padding:1rem 1.5rem;margin-bottom:0.8rem;border-radius:12px;border:none;font-weight:600;transition:all 0.3s ease;box-shadow:0 4px 6px rgba(0,0,0,0.07);display:flex;justify-content:space-between;align-items:center;}

    .section-btn:hover{transform:translateY(-2px);box-shadow:0 6px 12px rgba(0,0,0,0.1);}

    .btn-summary{background:linear-gradient(135deg,#a8a6af 0%,#3182ce 100%);color:white;}

    .btn-products{background:linear-gradient(135deg,#48bb78 0%,#3f604e 100%);color:white;}

    .btn-customers{background:linear-gradient(135deg,#7f6450 0%,#237863 100%);color:white;}

    .btn-category{background:linear-gradient(135deg,#7e5ac6 0%,#66597f 100%);color:white;}

    .card{border-radius:12px;box-shadow:0 4px 8px rgba(0,0,0,0.06);border:none;margin-top:0.5rem;}
```

```

.action-btn{border-radius:8px;padding:0.6rem 1.2rem;font-weight:600;transition:all 0.3s
ease;margin:0 0.5rem;}

.action-btn:hover{transform:translateY(-2px);}

.footer{color:#718096;font-size:0.9rem;margin-top:2rem;padding-top:1rem;border-top:1px solid
rgba(0,0,0,0.05);}

</style>

</head>

<body>

<div class="container">

<h2><i class="fas fa-chart-line me-2"></i>Analysis for <span class="filename">{{ filename
}}</span></h2>

<% with messages = get_flashed_messages() %>
<% if messages %>
<div class="alert alert-warning shadow-sm mb-4">{{ messages[0] }}</div>
<% endif %>
<% endwith %>

<button class="section-btn btn-summary" type="button" data-bs-toggle="collapse" data-bs-
target="#summaryCollapse">
<span><i class="fas fa-chart-pie me-2"></i>Summary Statistics</span> <i class="fas fa-
chevron-down"></i>
</button>
<div class="collapse" id="summaryCollapse"><div class="card card-body">{{ data.summary | 
safe }}</div></div>

<button class="section-btn btn-products" type="button" data-bs-toggle="collapse" data-bs-
target="#productsCollapse">
<span><i class="fas fa-shopping-bag me-2"></i>Top Products</span> <i class="fas fa-chevron-
down"></i>
</button>
<div class="collapse" id="productsCollapse"><div class="card card-body">{{ data.top_products | 
safe }}</div></div>

```

```

<button class="section-btn btn-customers" type="button" data-bs-toggle="collapse" data-bs-target="#customersCollapse">
    <span><i class="fas fa-users me-2"></i>Top Customers</span> <i class="fas fa-chevron-down"></i>
</button>

<div class="collapse" id="customersCollapse"><div class="card card-body">{{ data.top_customers | safe }}</div></div>

<button class="section-btn btn-category" type="button" data-bs-toggle="collapse" data-bs-target="#categoryCollapse">
    <span><i class="fas fa-tag me-2"></i>Revenue by Category</span> <i class="fas fa-chevron-down"></i>
</button>

<div class="collapse" id="categoryCollapse"><div class="card card-body">{{ data.category_revenue | safe }}</div></div>

<div class="d-flex justify-content-center mt-4">
    <a href="{{ url_for('dashboard') }}" class="btn btn-outline-primary action-btn"><i class="fas fa-arrow-left me-2"></i>Back to Dashboard</a>
    <a href="{{ url_for('upload') }}" class="btn btn-success action-btn"><i class="fas fa-plus me-2"></i>Upload Another File</a>
</div>

<div class="footer text-center">
    Built by Priya • Powered by Flask & Pandas
</div>
</div>
</body>
</html>

```

Sales of the Product Analysis of transactions.csv

 Summary Statistics 

 Top Products 

 Top Customers 

 Revenue by Category 

[!\[\]\(5f2ad55541d1c76614ad618336f6fa7b_img.jpg\) Back to Dashboard](#)

[!\[\]\(8290a0da7deb95092be3bf85b3086057_img.jpg\) Upload Another File](#)

Built by Priya • Powered by Flask & Pandas

style.css

```
/* Enhanced Button Styling */
```

```
.center-align {  
    text-align: center;  
}
```

```
.btn {  
    border-radius: 10px;  
    padding: 0.6rem 1.2rem;  
    font-weight: 500;  
    transition: all 0.3s ease;  
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.05);  
    background-color: white;  
    border: 2px solid transparent;  
}
```

```
.btn-outline-info {  
    border-color: #0dcaf0;  
    color: #0dcaf0;  
}  
  
.btn-outline-info:hover {  
    background: linear-gradient(135deg, #0dcaf0 0%, #31d2f2 100%);  
    color: white;  
    box-shadow: 0 6px 14px rgba(13, 202, 240, 0.3);  
    transform: translateY(-2px);  
}  
  
.btn-outline-secondary {  
    border-color: #6c757d;  
    color: #6c757d;  
}  
  
.btn-outline-secondary:hover {  
    background: linear-gradient(135deg, #6c757d 0%, #adb5bd 100%);  
    color: white;  
    box-shadow: 0 6px 14px rgba(108, 117, 125, 0.3);  
    transform: translateY(-2px);  
}  
  
.btn-outline-primary {  
    border-color: #0d6efd;  
    color: #0d6efd;  
}  
  
.btn-outline-primary:hover {
```

```

background: linear-gradient(135deg, #0d6efd 0%, #3d8bfd 100%);
color: white;
box-shadow: 0 6px 14px rgba(13, 110, 253, 0.3);
transform: translateY(-2px);
}

.btn-outline-success {
border-color: #198754;
color: #198754;
}

.btn-outline-success:hover {
background: linear-gradient(135deg, #198754 0%, #20c997 100%);
color: white;
box-shadow: 0 6px 14px rgba(25, 135, 84, 0.3);
transform: translateY(-2px);
}

```

Transaction.csv(saved in your file explorer)

TransactionID, CustomerID, ProductID, ProductCategory, Quantity, Price, TransactionDate

T001,C101,P501,Electronics,1,500,2023-01-05

T002,C102,P601,Grocery,5,20,2023-01-06

T003,C101,P701,Clothing,2,150,2023-01-06

T004,C103,P601,Grocery,10,20,2023-01-07

T005,C104,P801,Electronics,1,1200,2023-01-07

T006,C102,P701,Clothing,3,150,2023-01-08

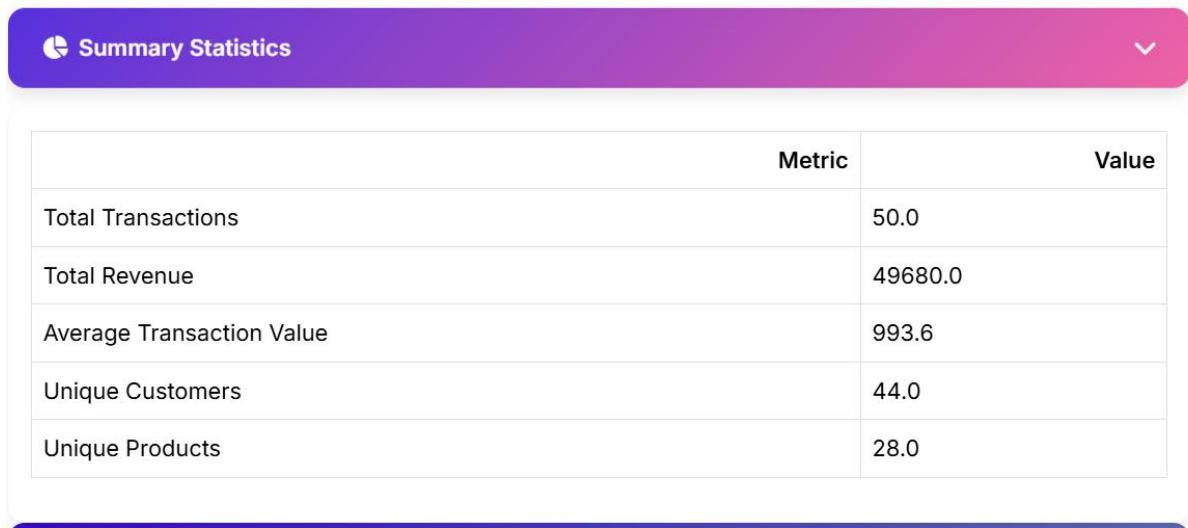
T007,C105,P901,Furniture,1,2000,2023-01-09

T008,C101,P601,Grocery,2,20,2023-01-09
T009,C106,P701,Clothing,4,150,2023-01-10
T010,C102,P801,Electronics,1,1200,2023-01-10
T011,C107,P601,Grocery,8,20,2023-01-11
T012,C108,P502,Electronics,2,750,2023-01-12
T013,C109,P702,Clothing,1,200,2023-01-12
T014,C110,P903,Furniture,1,2500,2023-01-13
T015,C101,P601,Grocery,6,20,2023-01-13
T016,C111,P802,Electronics,1,1100,2023-01-14
T017,C112,P701,Clothing,5,150,2023-01-14
T018,C113,P601,Grocery,3,20,2023-01-15
T019,C114,P904,Furniture,2,1800,2023-01-15
T020,C115,P503,Electronics,1,950,2023-01-16
T021,C102,P702,Clothing,2,200,2023-01-16
T022,C116,P601,Grocery,4,20,2023-01-17
T023,C117,P701,Clothing,1,150,2023-01-17
T024,C118,P803,Electronics,3,600,2023-01-18
T025,C119,P905,Furniture,1,2200,2023-01-18
T026,C120,P601,Grocery,7,20,2023-01-19
T027,C121,P703,Clothing,2,175,2023-01-19
T028,C122,P804,Electronics,1,1300,2023-01-20
T029,C123,P906,Furniture,1,2100,2023-01-20
T030,C124,P601,Grocery,5,20,2023-01-21
T031,C125,P701,Clothing,2,150,2023-01-21
T032,C126,P805,Electronics,2,850,2023-01-22
T033,C127,P907,Furniture,1,2400,2023-01-22
T034,C128,P601,Grocery,9,20,2023-01-23
T035,C129,P704,Clothing,3,160,2023-01-23
T036,C130,P806,Electronics,1,1400,2023-01-24
T037,C131,P908,Furniture,2,1950,2023-01-24
T038,C132,P601,Grocery,4,20,2023-01-25

T039,C133,P701,Clothing,6,150,2023-01-25
T040,C134,P807,Electronics,1,1250,2023-01-26
T041,C135,P909,Furniture,1,2300,2023-01-26
T042,C136,P601,Grocery,3,20,2023-01-27
T043,C137,P705,Clothing,2,180,2023-01-27
T044,C138,P808,Electronics,2,950,2023-01-28
T045,C139,P910,Furniture,1,2600,2023-01-28
T046,C140,P601,Grocery,5,20,2023-01-29
T047,C141,P701,Clothing,1,150,2023-01-29
T048,C142,P809,Electronics,1,1350,2023-01-30
T049,C143,P911,Furniture,1,2000,2023-01-30
T050,C144,P601,Grocery,6,20,2023-01-31

10. Screenshots & Output Explanation

- **Top Products:** Shows the most purchased items
- **Top Customers:** Highlights high spenders
- **Category Revenue:** Reveals revenue distribution
- **Summary Statistics:** Includes average transaction value and purchase frequency



Metric	Value
Total Transactions	50.0
Total Revenue	49680.0
Average Transaction Value	993.6
Unique Customers	44.0
Unique Products	28.0

Top Products



ProductID	Quantity
P601	77
P701	24
P702	3
P704	3
P803	3

Top Customers



CustomerID	TotalAmount
C131	3900
C114	3600
C139	2600
C110	2500
C127	2400

*

Revenue by Category

ProductCategory	TotalAmount
Furniture	25600
Electronics	17150
Clothing	5390
Grocery	1540

[← Back to Dashboard](#) [+ Upload Another File](#)

127.0.0.1:5000/result/transactions.csv

Sales of the Product Analysis of transactions.csv

- Summary Statistics
- Top Products
- Top Customers
- Revenue by Category

[← Back to Dashboard](#) [+ Upload Another File](#)

Built by Priya • Powered by Flask & Pandas

11. Closure & Bibliography

This project showcases the power of Pandas for business analytics without relying on visualizations. It emphasizes clean code, reproducibility, and modular design.

References:

- [Pandas Documentation](#)
 - [Python Official Docs](#)
-