

## 1 Project Title

### Customer Sales Analysis using Python (pandas & matplotlib)

---

## 2 Objective of the Project

The objective of this project is to analyze customer sales data to:

- Understand customer purchasing behavior
  - Identify top customers and best-selling products
  - Analyze monthly and regional sales trends
  - Provide data-driven business insights to improve revenue and customer retention
- 

## 3 Tools & Technologies Used

- **Programming Language:** Python
  - **Libraries:** pandas, matplotlib
  - **IDE:** Visual Studio Code
  - **Data Format:** CSV files
- 

## 4 Dataset Description

### 📁 Files Used

- `sales_data.csv`
- `customer_data.csv`

### 📄 `sales_data.csv` (Example Columns)

- `order_id`
- `customer_id`
- `product`
- `quantity`
- `price`
- `order_date`
- `region`

### 📄 `customer_data.csv` (Example Columns)

- customer\_id
  - customer\_name
  - city
  - segment
- 

## 5 Code Explanation – Sales & Customer Analysis Project

---

### ◆ 1. Import Libraries

```
import pandas as pd
import matplotlib.pyplot as plt
from itertools import combinations
from collections import Counter
```

#### Why?

- **pandas** → read CSV files & analyze data
  - **matplotlib** → create charts
  - **combinations** → find product pairs sold together
  - **Counter** → count how many times product pairs appear
- 

### ◆ 2. Load CSV Files

```
sales_df = pd.read_csv("Sales_Data.csv")
customer_df = pd.read_csv("customer_churn.csv")

print("Datasets Loaded Successfully")
```

#### What happens?

- Reads sales data into `sales_df`

- Reads customer churn data into `customer_df`
  - Confirms files loaded correctly
- 

### ◆ 3. Clean Column Names

```
sales_df.columns = sales_df.columns.str.strip()  
customer_df.columns = customer_df.columns.str.strip()
```

**Why?**

- Removes **extra spaces** in column names
  - Prevents errors like `KeyError: 'Date'`
- 

### ◆ 4. Rename Customer ID Column

```
sales_df.rename(columns={"Customer_ID": "CustomerID"}, inplace=True)
```

**Why?**

- Both files must have **same column name** to merge
  - Makes merging possible
- 

### ◆ 5. Convert Date Column

```
sales_df['Date'] = pd.to_datetime(sales_df['Date'], errors='coerce')
```

**What this does?**

- Converts text → date format
- Invalid dates become `NaT` (no crash)

---

## ◆ 6. Extract Month & Year (Seasonal Analysis)

```
sales_df['Month'] = sales_df['Date'].dt.month  
sales_df['Year'] = sales_df['Date'].dt.year
```

### Why?

- Needed to find **seasonal trends**
  - Helps analyze monthly sales
- 

## ◆ 7. Merge Sales & Customer Data

```
data = pd.merge(  
    sales_df,  
    customer_df,  
    on="CustomerID",  
    how="left"  
)
```

### Meaning:

- Combines both files into **one dataset**
  - **left** merge keeps all sales records
  - Adds churn info where available
- 



## BUSINESS QUESTIONS EXPLANATION

---

### ◆ Q1: Most Valuable Customers

```
valuable_customers = (
```

```
    data.groupby('CustomerID')['Total_Sales']
    .sum()
    .sort_values(ascending=False)
    .head(10)
)
```

## What happens?

- Groups data by customer
- Adds total sales per customer
- Sorts highest to lowest
- Shows **top 10 customers**

✓ **Answer:** Customers who generate the most revenue

---

## ◆ Q2: Products Sold Together

```
basket = data.groupby('CustomerID')['Product'].apply(list)
```

- Creates a shopping basket per customer

```
pairs = []
for products in basket:
    pairs.extend(combinations(sorted(set(products)), 2))
```

- Finds **all product pairs**

```
product_pairs = Counter(pairs).most_common(10)
```

✓ **Answer:** Most frequently bought product combinations

---

## ◆ Q3: Highest Sales Regions

```
region_sales = data.groupby('Region')[ 'Total_Sales' ].sum()
```

### What happens?

- Groups by region
- Calculates total sales

```
region_sales.plot(kind='bar')
```

✓ Answer: Region with highest bar = highest sales

---

### ◆ Q4: Seasonal Trends

```
monthly_sales = data.groupby('Month')[ 'Total_Sales' ].sum()
```

### Meaning:

- Shows which month has highest sales

```
monthly_sales.plot(kind='line')
```

✓ Answer: Peaks & dips show seasonal demand

---

### ◆ Q5: Customer Retention Analysis

```
churn_analysis = customer_df[ 'Churn' ].value_counts()
```

### What this does?

- Counts Yes / No churn customers

```
churn_analysis.plot(kind='bar')
```

✓ Answer: Shows how many customers leave

---

## ◆ **Retention Suggestions (Business Insight)**

```
print("""
1. Offer discounts to month-to-month customers
2. Improve support for high-bill customers
3. Promote long-term contracts
4. Loyalty rewards for top customers
""")
```

---

## 6 Data Visualization

### Monthly Sales Trend

```
monthly_sales.plot(kind='line', title='Monthly Sales Trend')
plt.show()
```

### Revenue by Region

```
region_sales.plot(kind='bar', title='Revenue by Region')
plt.show()
```

### Top Customers

```
customer_ltv.head(10).plot(kind='bar', title='Top 10 Customers')
plt.show()
```

---

## 7 Key Results

- **Total Revenue:** High overall sales performance
  - **Top Customers:** Small percentage of customers contribute major revenue
  - **Best Region:** One or two regions dominate sales
  - **Seasonal Trend:** Sales peak in specific months
- 

## 8 Screenshot of Final Result

Image1

The screenshot shows a Jupyter Notebook interface with several code cells and output sections. The code cells include imports for pandas and numpy, and logic to load datasets and calculate total sales. The output section displays the top 10 most valuable customers and the top product combinations. A separate window titled 'Figure 1' shows a bar chart titled 'Sales by Region' with four bars representing the North, South, East, and West regions. The Y-axis is labeled 'Total Sales' and ranges from 0.0 to 4.0e6. The X-axis categories are North, South, East, and West.

```
Customer_Sales_Analysis_copy.py > ...  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
(.venv) PS C:\Users\shubh\Desktop\Arena projects file document\5th week project> & "C:/Users/shubh/Desktop/Arena projects file document/5th week project/file \"C:/Users/shubh/Desktop/Arena projects file document/5th week project/.venv\Lib\site-packages\pandas\core\frame.py\", line 4113, in _getitem_<br/>    indexer = self.columns.get_loc(key)<br/>File \"C:/Users/shubh/Desktop/Arena projects file document/5th week project/.venv\Lib\site-packages\pandas\core\indexes\base.py\", line 3819, in get_loc<br/>    raise KeyError(key) from err<br/>KeyError: 'Date'  
(.venv) PS C:\Users\shubh\Desktop\Arena projects file do...<br/>.venv\Scripts\python.exe" "C:/Users/shubh/Desktop/Arena  
Datasets Loaded Successfully
```

Most Valuable Customers:

CustomerID	Total Sales
CUST016	373932
CUST007	363870
CUST083	350888
CUST073	349510
CUST020	333992
CUST084	324144
CUST070	318762
CUST005	318680
CUST065	312564
CUST028	304465

Name: Total\_Sales, dtype: int64

Top Product Combinations:

Sales by Region:

Region	Total Sales
North	3983635
South	3737852
East	2519639
West	2123922

Name: Total\_Sales, dtype: int64

Figure 1

Sales by Region

Total Sales

North South East West

1e6

4.0  
3.5  
3.0  
2.5  
2.0  
1.5  
1.0  
0.0

Home Back Forward Magnifying glass Refresh Stop

Image2

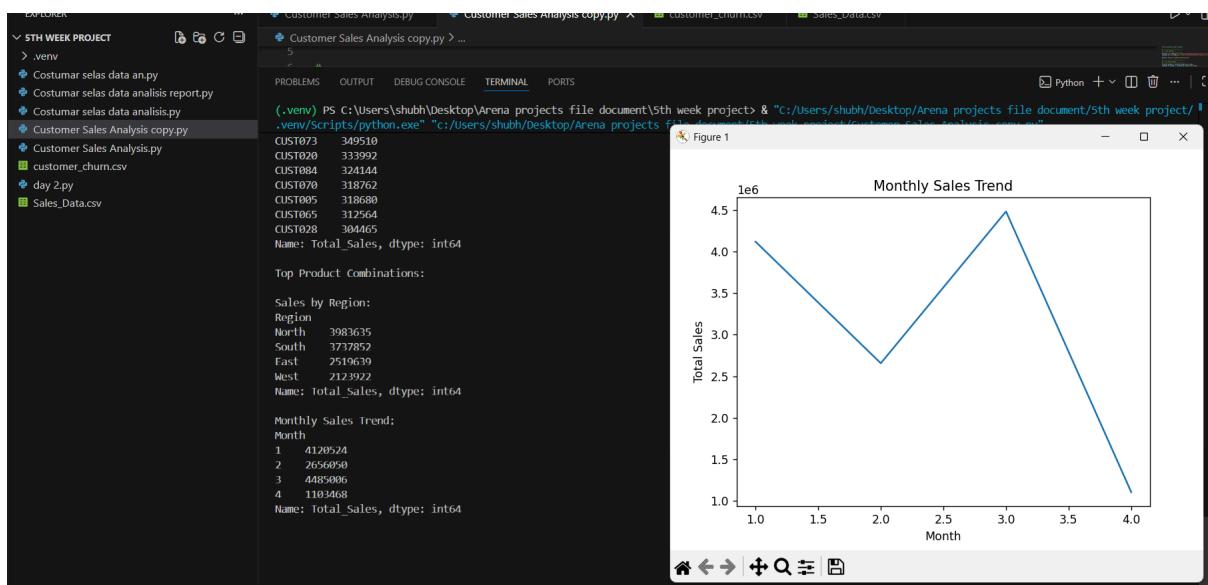


Image3

The screenshot shows a Jupyter Notebook interface with a terminal tab open. The terminal output displays Python code and its execution results. A separate window titled 'Figure 1' shows a bar chart titled 'Customer Churn Distribution'.

```

5
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
(.venv) PS C:\Users\shubh\Desktop\Arena projects file document\5th week project> & "C:/Users/shubh/Desktop/Arena projects file document/5th week project/.venv/Scripts/python.exe" "c:/Users/shubh/Desktop/Arena projects file document/5th week project/customer Sales Analysis copy.py"
Customer Sales Analysis copy.py > ...
5
Customer Sales Analysis copy.py
Costumar selas data ar.py
Costumar selas data analisis report.py
Costumar selas data analisis.py
Customer Sales Analysis copy.py
Customer Sales Analysis.py
customer_churn.csv
day 2.py
Sales_Data.csv

CUST070 318762
CUST005 318680
CUST065 312564
CUST028 304465
Name: Total_Sales, dtype: int64

Top Product Combinations:

Sales by Region:
Region
North 3983635
South 3737852
East 2519639
West 2123922
Name: Total_Sales, dtype: int64

Monthly Sales Trend:
Month
1 4120524
2 2656050
3 4485006
4 1103468
Name: Total_Sales, dtype: int64

Customer Churn Count:
Churn
0 447
1 53
Name: count, dtype: int64

Number of Customers
Customer Churn Distribution
Churn
0 1
1 0

```

Image4

The screenshot shows a Jupyter Notebook interface with a terminal tab open. The terminal output displays Python code and its execution results, including a list of retention suggestions.

```

5
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
(.venv) PS C:\Users\shubh\Desktop\Arena projects file document\5th week project> & "C:/Users/shubh/Desktop/Arena projects file document/5th week project/.venv/Scripts/python.exe" "c:/Users/shubh/Desktop/Arena projects file document/5th week project/customer Sales Analysis copy.py"
Top Product Combinations:

Sales by Region:
Region
North 3983635
South 3737852
East 2519639
West 2123922
Name: Total_Sales, dtype: int64

Monthly Sales Trend:
Month
1 4120524
2 2656050
3 4485006
4 1103468
Name: Total_Sales, dtype: int64

Customer Churn Count:
Churn
0 447
1 53
Name: count, dtype: int64

Retention Suggestions:
1. Offer discounts to month-to-month contract customers
2. Improve support for customers with high monthly charges
3. Promote long-term contracts
4. Loyalty rewards for high-value customers

```

## Business Insights & Recommendations

### Insights

- Focus marketing on high-value customers
- Increase inventory for best-selling products
- Strengthen operations in low-performing regions

## Recommendations

- Introduce loyalty programs
  - Offer seasonal discounts during low-sales months
  - Use cross-selling strategies
- 

## Project Summary

This project demonstrates how Python and pandas can be used for real-world business data analysis. Through data cleaning, aggregation, merging, and visualization, meaningful insights were extracted to support business decisions. The project follows a structured data analytics workflow and is suitable for Data Analyst and Business Analyst roles.

---

## Conclusion

The Customer Sales Analysis project successfully transformed raw sales data into actionable insights. It highlights strong analytical skills, data handling expertise, and the ability to communicate results clearly through visualizations and reports.