AURORA DEEMED TO BE UNIVERSITY
AURORA HIGHER EDUCATION AND RESEARCH ACADEMY

aurora™ TEMPLE OF LEARNING

Course Name: Python Programming

Course Code: CS201

2nd Year & 1st Term , B.Tech (AIML – IIC)

Title: Sales Data Aggregator

Faculty Name: Ms. Jayasri K

A. Vignesh (241U1R2089)

Content

- Abstract
- Introduction
- Problem Statement
- Literature Survey
- Objectives
- Methodology
- Tools & Technologies Used
- Dataset Description
- Implementation
- Output & Results
- Key Learnings
- Applications
- Future enhancement
- Conclusion

Abstract

In today's competitive business environment, efficient management and analysis of sales data are crucial for informed decision-making. This research presents the development of a Sales Data Aggregator, a Python-based system designed to collect, clean, and consolidate sales information from multiple sources. The system leverages data processing and visualization techniques to identify trends, monitor performance, and generate actionable insights. By automating data aggregation and analysis, the project reduces manual effort, improves accuracy, and enables businesses to make timely, data-driven decisions.

Introduction

What is Sales Data?

- Records of what was sold, how much, at what price, where, and when.
- Acts as the foundation for tracking a company's performance.

Why Analyze Sales Data?

- Understand customer demand and seasonal trends.
- Compare performance across stores or regions.

Problem Statement

Current Business Challenges:

- Data is scattered across multiple sources (Excel, CSV, databases).
- Column names and formats are inconsistent across files.
- Lack of timely insights delays decision-making.

Impact:

- Businesses cannot track sales trends quickly.
- Managers spend more time cleaning data than analyzing it.

Literature Survey

Article	Focus	Gaps Identified
1. Monthly Sales Commission Analysis (Durga Gadiraju)	Shows how to clean, merge, and aggregate sales data for commission calculation using groupby()	Focus is only on commission; does not provide a general tool for overall sales aggregation or flexible reporting.
2. Data Analysis using Python (AnalytixLabs)	Covers full workflow: importing, cleaning, aggregating, and visualizing sales data. Useful for business insights.	Workflow is totally manual and not digital; lacks automation and dynamic dashboards for repeated analysis.
3. Pandas: Data Aggregation & Grouping (Ebrahim Mousavi)	Explains "split-apply-combine" method and multi-level grouping for detailed analysis.	Technical explanation only; results are just in text format and not in visualizations
4. Data Aggregation & Grouping (Aman Kharwal)	Beginner guide to functions like sum(), mean(), count() and handling missing values.	Too basic for complex sales needs; does not cover large datasets, automation, or advanced insights.

Objectives

1.Collect sales data from multiple files and formats into one place.

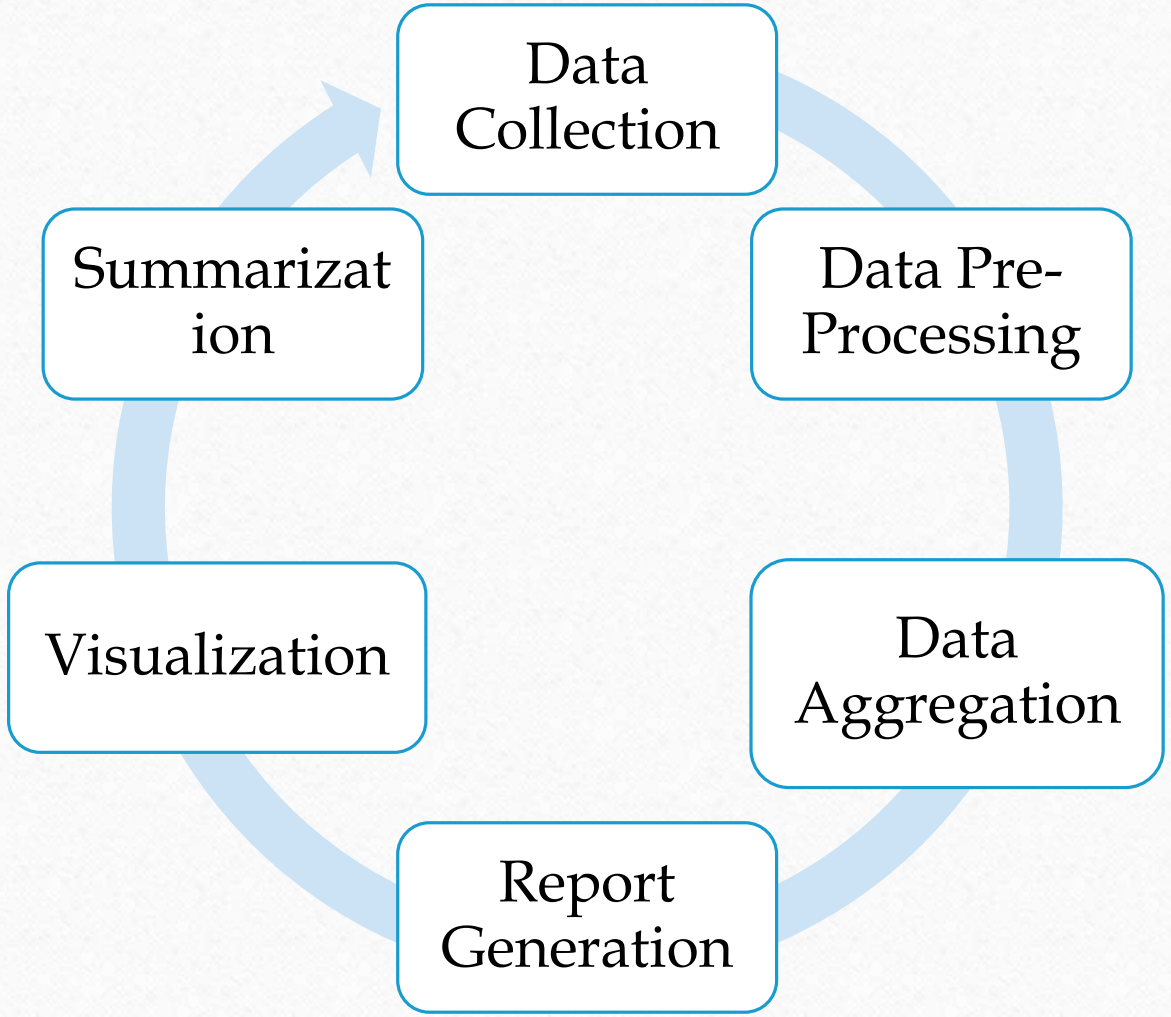
2.Clean and standardize raw data to remove errors and inconsistencies.

3.Aggregate and summarize sales by product, region, and time period.

4.Visualize insights using charts and graphs for easy understanding.

5.Generate reports that support quick and accurate business decisions.

Methodology



Tools and Technologies Used

Programming Language:

- Python 3.13.5

Data Formats:

- .csv and .xlsx supported as input.

Libraries:

- **os** → File path handling, saving uploaded files, and directory management
- **pandas** → Data manipulation and cleaning.

Tools and Technologies Used...

- **matplotlib** → Creating plots like bar charts, line charts, and custom visualizations
- **seaborn** → High-level statistical plots with professional styling
- **flask** → Web framework for handling routes, requests, and rendering templates
- **werkzeug.utils (secure_filename)** → Safely handling uploaded file names to avoid conflicts or security risks.

Code in Separate File

Dataset Description

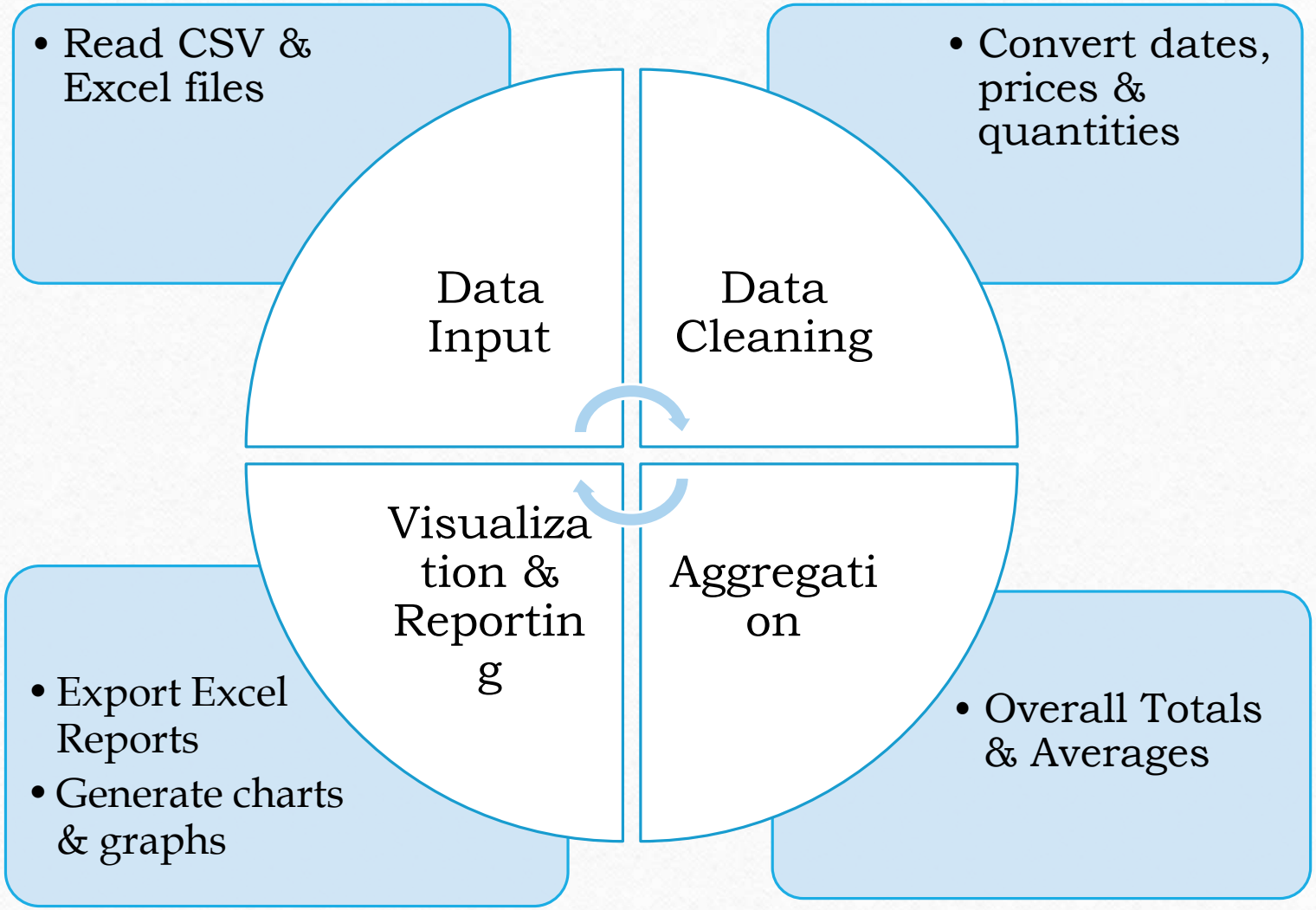
Attributes:

- Date / Order Date / Invoice Date – transaction time.
- Store / Region / Retailer – where the sale occurred.
- Product / Product Name – item sold.
- Quantity / Units Sold – amount purchased.
- Price / Total Sales – cost per unit and total sales value.

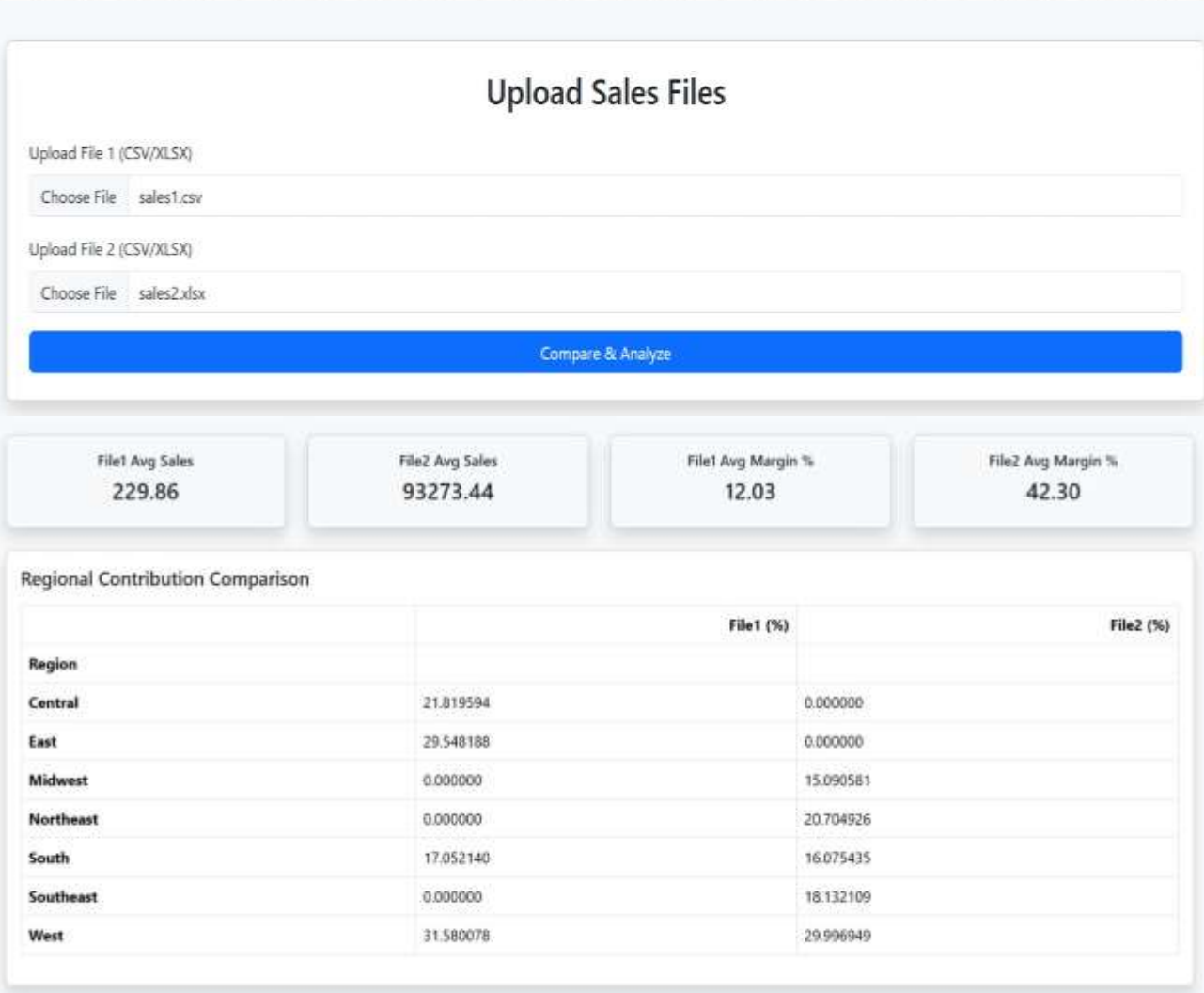
Challenges in Dataset:

- Different naming conventions (e.g., “Store” vs. “Region”).
- Missing values.
- Different encoding types (UTF-8, Latin1).

Implementation

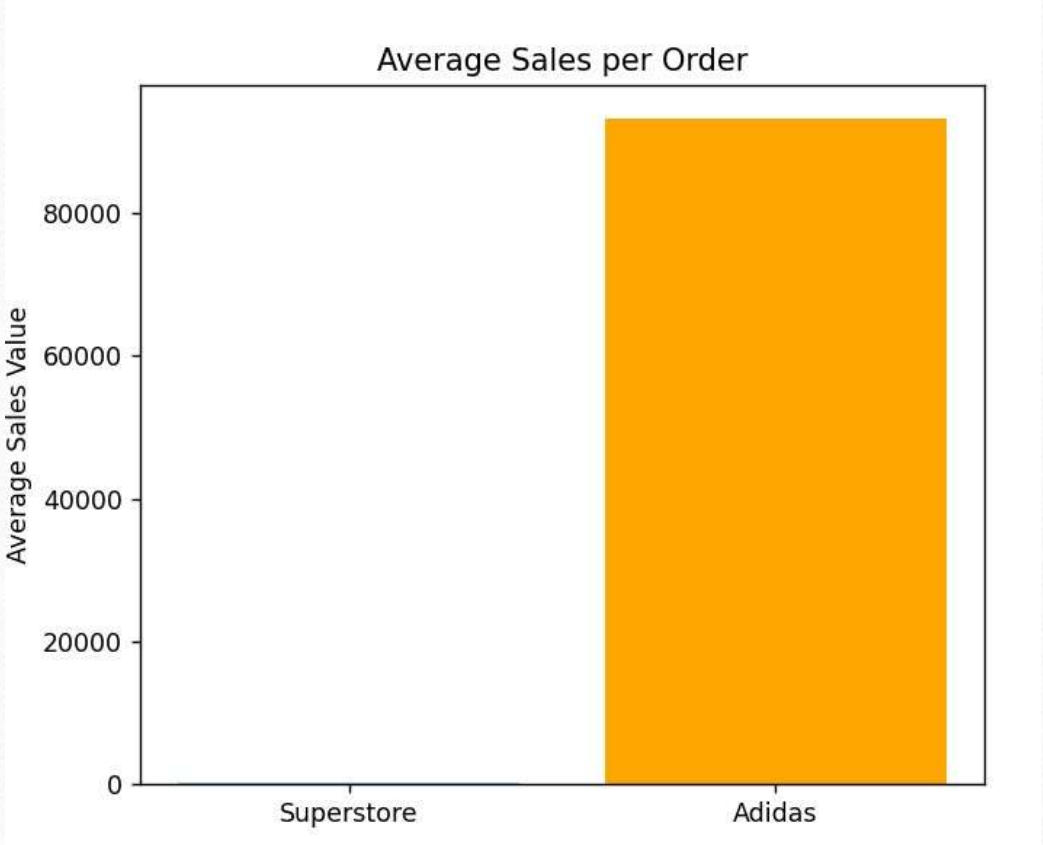


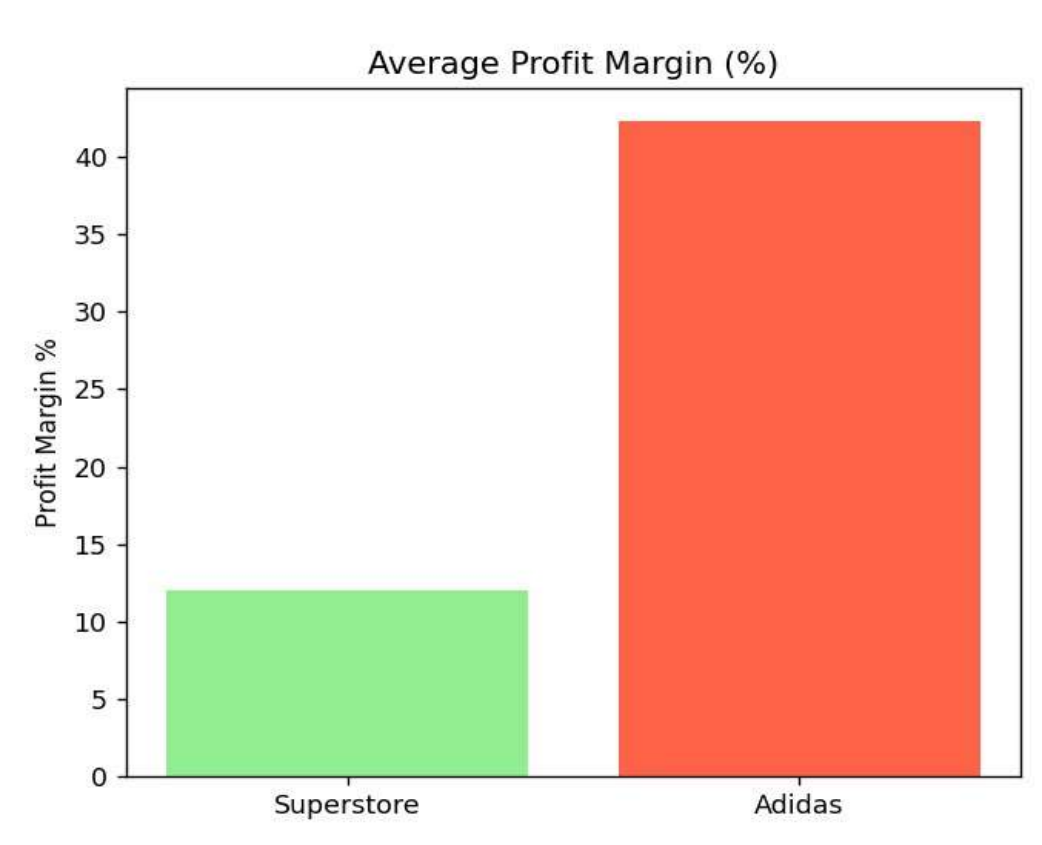
Outputs & Results



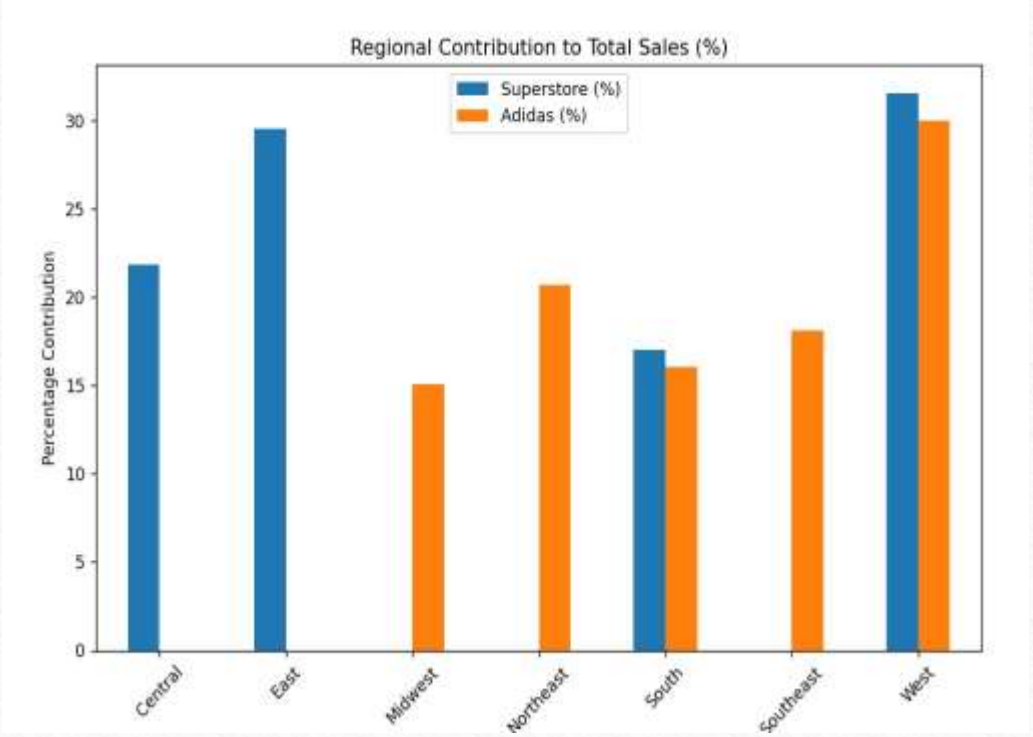
Region	File1 (%)	File2 (%)
Central	21.819594	0.000000
East	29.548188	0.000000
Midwest	0.000000	15.090581
Northeast	0.000000	20.704926
South	17.052140	16.075435
Southeast	0.000000	18.132109
West	31.580078	29.996949

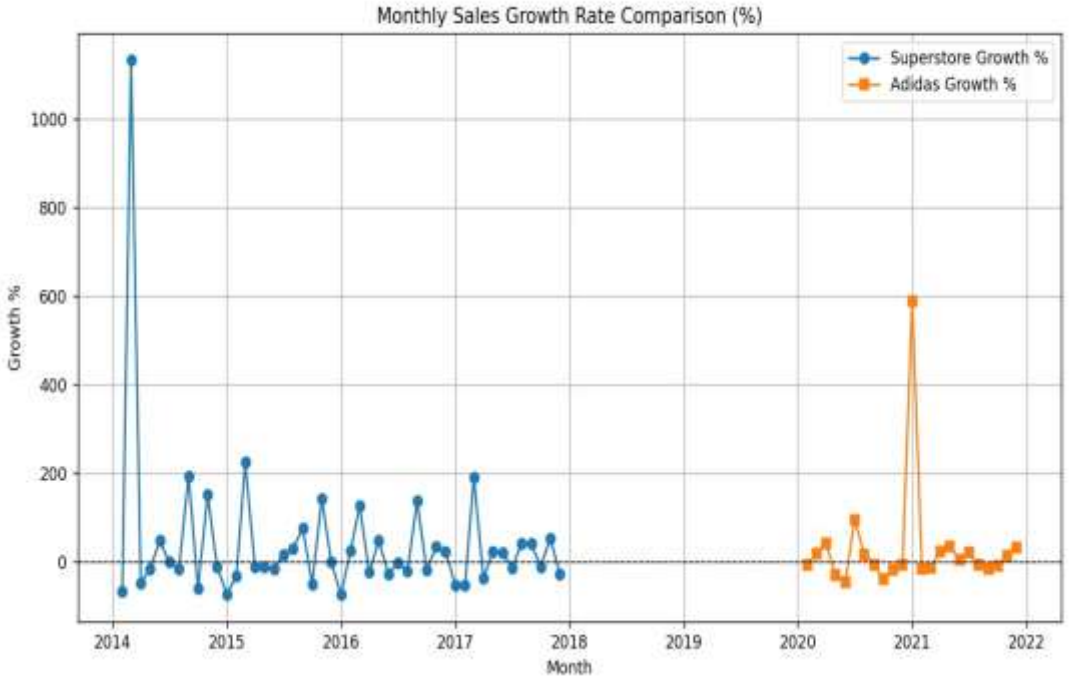
Outputs & Results...



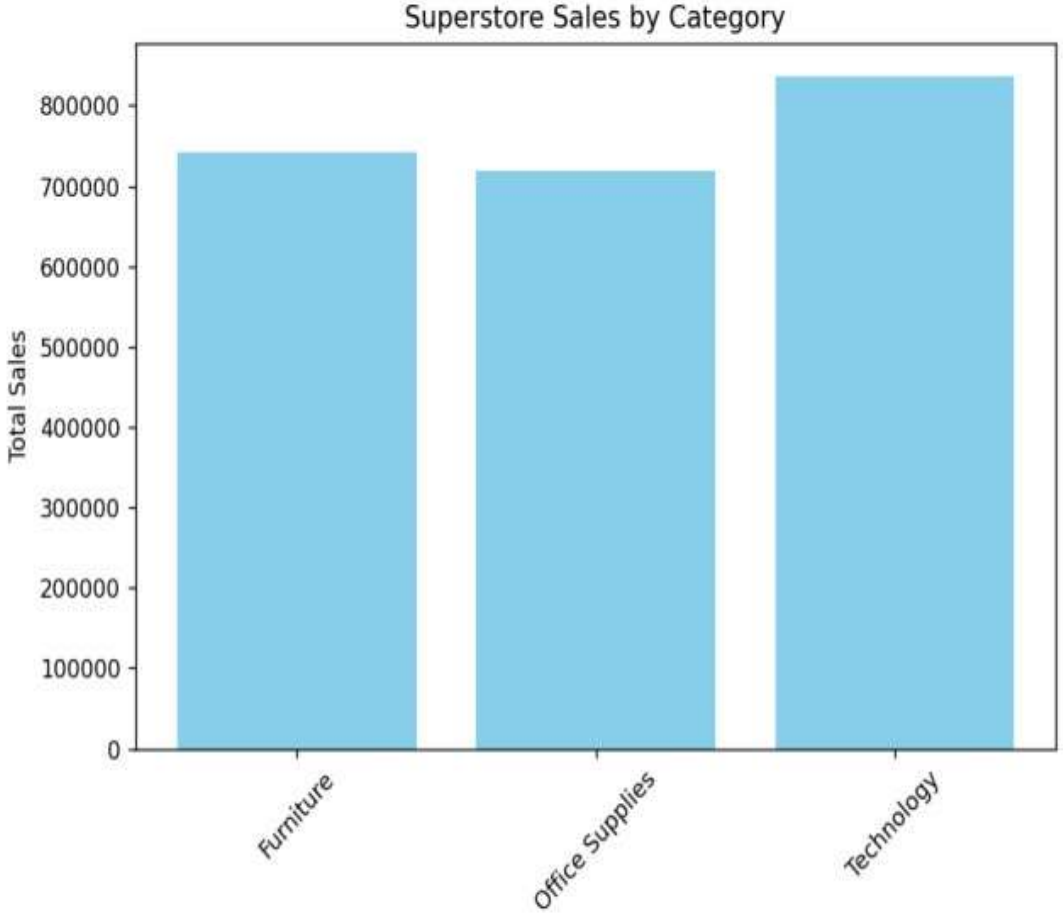


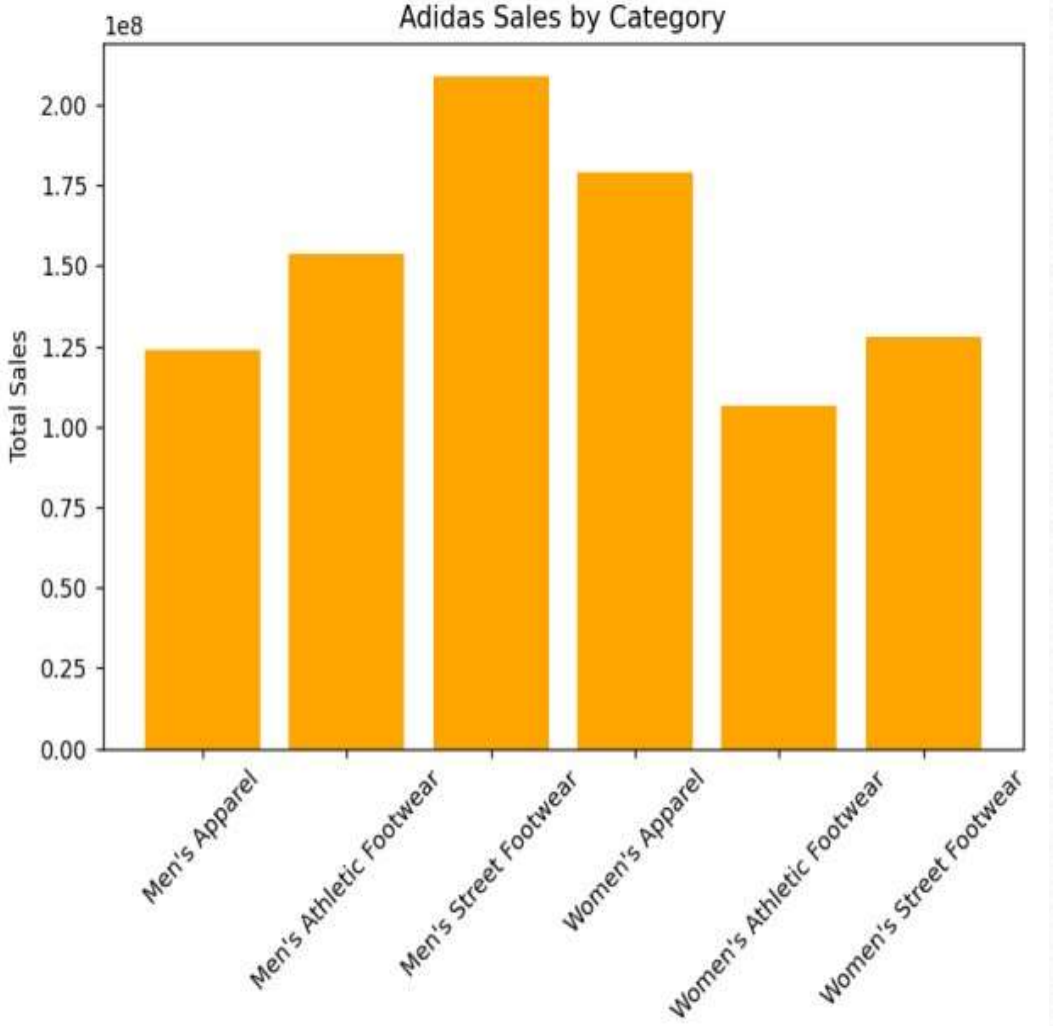
Outputs & Results...



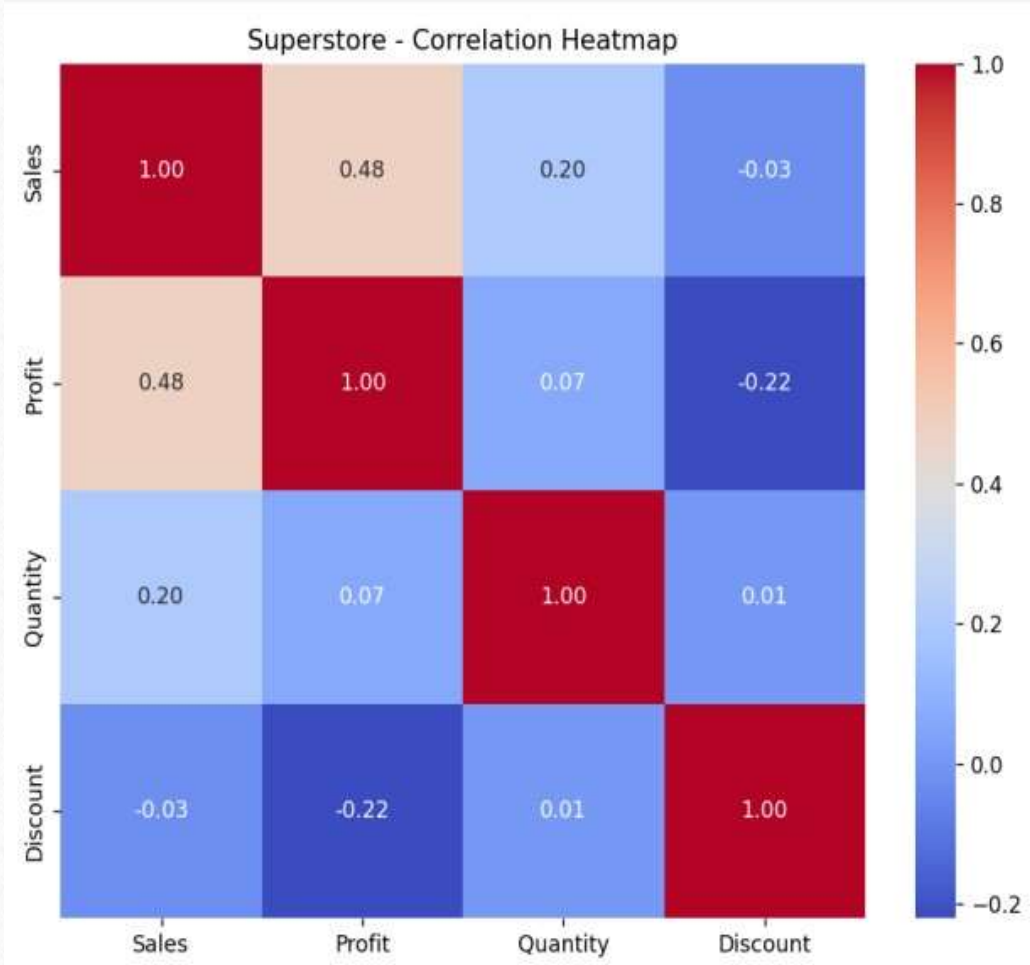


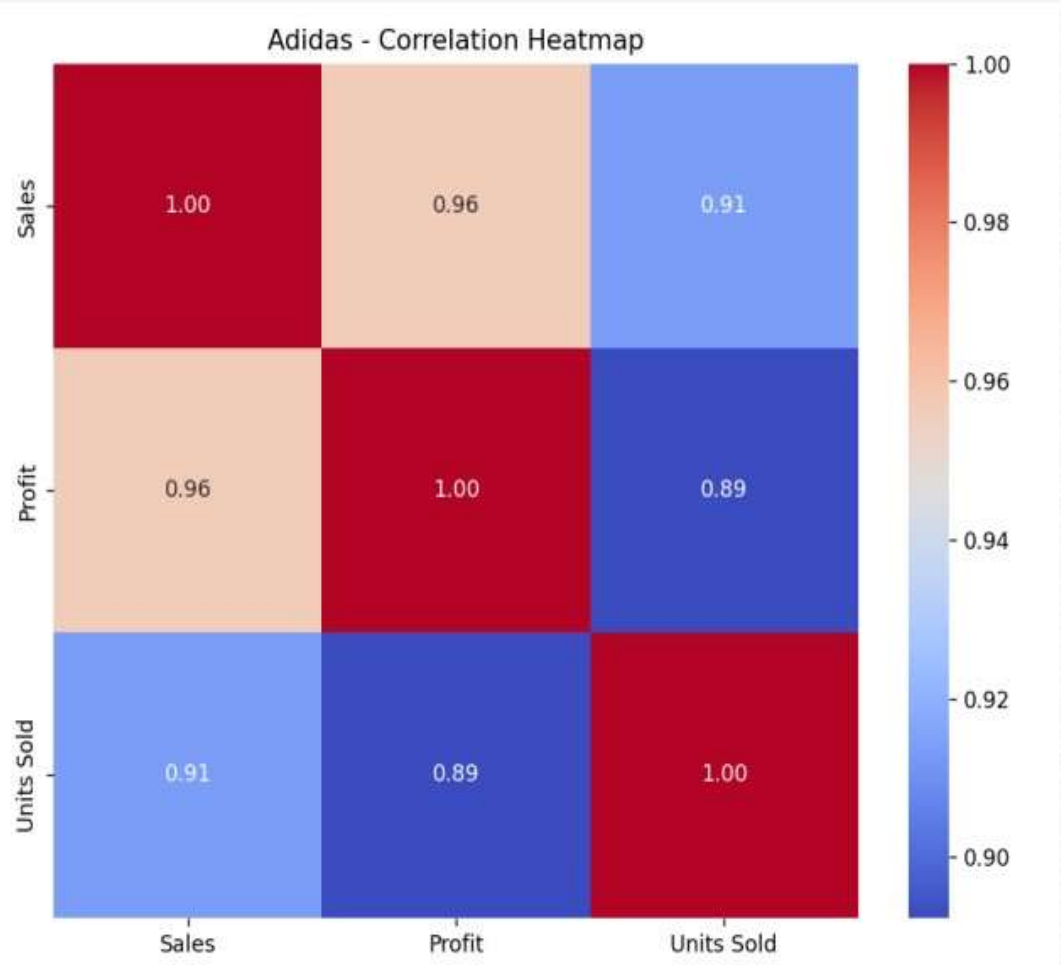
Outputs & Results...





Outputs & Results...





Discussion

This analysis reveals two distinct business models: a traditional high-volume retailer versus a high-value, brand-focused powerhouse.

Superstore: The Volume Game

- Business Model: Relies on high volume with a low average sale of $230.
- Profitability: Operates on slim margins (12%) that are negatively impacted by discounts.
- Growth: Stable and consistent, but low.

Adidas: The Value Game

- Business Model: Drives massive value with a high average sale of $93,000.
- Profitability: Commands an exceptional profit margin of 42%. Sales, profit, and units sold are strongly correlated.
- Growth: Dynamic and volatile, with huge event-driven spikes.

Key Learnings

Technical:

- Importance of handling different file formats and encodings.
- Using Pandas for powerful data manipulation.
- Generating professional reports programmatically.

Conceptual:

- Data cleaning is often 70% of the work in real projects.
- Visuals communicate insights much faster than raw numbers.
- Modular design makes code reusable for other datasets.

Applications

1.Retail Chains – Compare sales across multiple stores to find top and low performers.

2.E-Commerce Platforms – Track product demand, seasonal trends, and customer preferences.

3.FMCG Companies – Monitor region-wise sales to optimize distribution and supply chain.

4.Small & Medium Businesses – Automatically generate sales reports, saving time and effort.

5.Franchise Management – Consolidate and standardize data.

Future Enhancements

1. **Real-Time Data Integration** – Connect with live databases and APIs for instant updates.
2. **Machine Learning Insights** – Predict future sales trends using ML models.
3. **Multi-Format Support** – Extend beyond CSV/Excel to JSON, SQL, and cloud sources.
4. **User Authentication** – Provide secure, role-based access to reports.

Conclusion

The Sales Data Aggregator makes sales analysis easier by cleaning, combining, and summarizing data automatically. It saves time, reduces errors, and helps businesses understand their performance better. In the future, it can grow into a smarter tool with real-time updates and advanced insights.

Thank You