

SAMSUNG PROJECT REPORT

King's Coffee Shop Analytics

- A Data-Driven Decision Support System

Developed By:

H Ranjitha –(20231CSD0114)

B-tech -Computer Science & Engineering (Data Science Specialization)

Presidency University, Bengaluru

Date of Submission:

July 17, 2025

Course: Samsung Innovation Campus

Subject: Coding and Programming in Python

INDEX

- 1. Problem Statement**
- 2. Project Description**
- 3. Data Overview & Output Information**
- 4. Solution Plan**
- 5. Design**
- 6. Implementation**
- 7. Code & Explanation**
- 8. Output Screenshots**
- 9. Closure**
- 10. Bibliography**

Problem Statement

In today's fast-paced and data-driven business environment, local coffee shops often lack the tools and insights to make informed decisions that can boost profitability and improve customer satisfaction. *King's Coffee Shop*, a fictional but representative example of a mid-sized cafe, faces challenges in understanding its customer preferences, managing inventory effectively, optimizing profit margins, and identifying key sales trends.

Despite having access to basic sales and customer data, the shop lacks a centralized system to analyze this information and translate it into actionable strategies. Without proper analytics, the business risks overstocking or understocking, missing out on high-margin products, and failing to retain loyal customers.

This project aims to solve that problem by building a **Python-based Data Analytics Platform** that processes real-world style datasets and performs **descriptive, diagnostic, predictive, and prescriptive analytics**. By visualizing insights and offering CRUD functionalities, this tool empowers King's Coffee Shop to make smarter business decisions backed by data.

Project Description

The system provides a complete end-to-end solution for analyzing historical data related to **sales, inventory, profit margins, and customer behavior**. In addition to analytics, the project integrates a CRUD (Create, Read, Update, Delete) module that allows real-time manipulation of sales, inventory, customer, and margin records through a menu-driven interface.

Key features include:

- Monthly and item-wise **Sales Analysis**
- Forecasting **Inventory Requirements** based on recent trends
- Optimization of **Profit Margins** by analyzing cost and pricing data
- Identification of **Customer Behavior Patterns**, such as top spenders and return rates
- A dynamic **CRUD Console** to update and manage data interactively

The project structure has been carefully modularized with organized folders for **data, scripts, models, outputs, configuration, and visuals**, following best practices in software development. It has been designed keeping scalability and future enhancements in mind, such as transitioning to a GUI dashboard or database integration.

By simulating a real-world retail scenario, this project not only showcases technical proficiency in Python and analytics but also builds a strong foundation for understanding how data-driven decisions can improve business operations.

Data, Output Info, Purpose, Outcome & Benefits

Data Used in the Project

The project uses four key CSV files stored in the `kings_coffee_data/` folder, simulating real business datasets:

1. `sales_data.csv` – Contains date-wise sales transactions including item, quantity, cup size, and total_price.
2. `inventory_data.csv` – Includes current stock levels, reorder thresholds, and supplier-related info.
3. `customer_data.csv` – Logs customer visits with customer_id, visit_date, and amount spent.
4. `margin_data.csv` – Details product-level cost, selling price, and calculated margins.

These datasets were either synthetically generated or simulated using Indian-specific names, pricing (₹), and realistic patterns for training and evaluation.

Output Information

Each analytics module produces:

- Printed summaries in the console (e.g., monthly sales, top items, inventory alerts).
- Saved CSV files with analysis results in `csv_outputs/`
- Bar plots and charts stored in `screenshots_plots/` and shown on screen for insights.

Expected Outcomes

- Owners gain data-driven clarity on their shop's performance month by month.
- Understand top-selling items, peak sales periods, and low inventory risks.
- Identify returning customers and top spenders to enable loyalty campaigns.
- Analyze profit margins across products to optimize pricing.

Key Benefits

- Provides a single consolidated system for analytics and data management.
- Modular design helps in adding new datasets or analytics easily.
- Insights lead to cost-saving strategies, increased sales, and customer retention.
- Prepares the developer to work on real-world data engineering and analytics scenarios.

Solution Plan

To effectively analyze and improve the operations of King's Coffee Shop, I broke down the solution into the following clear, modular steps:

1. Define Problem Areas

Identify four major areas where data analytics can help the coffee shop:

- Sales trends and performance
 - Inventory forecasting and alerts
 - Profit margin optimization
 - Customer behavior and loyalty
-

2. Collect & Organize Data

Structure and simulate realistic CSV data files:

- sales_data.csv, inventory_data.csv, customer_data.csv, margin_data.csv
 - Ensure consistent formatting, dates, and Indian pricing (₹)
-

3. Create Modular Scripts

Divide the solution into modules:

- `sales_analysis.py` → Analyze monthly sales, top items, cup size popularity
 - `inventory_forecast.py` → Predict stock shortages, reorder needs
 - `margin_optimizer.py` → Find profitable items, low-margin alerts
 - `customer_behaviour.py` → Study repeat customers, top spenders, visit frequency
 - `crud_console.py` → CRUD operations to modify and update records
 - `main.py` → A single entry point with a console menu for the entire system
-

4. Visualize and Store Output

- Use matplotlib and seaborn for clear visualizations
 - Save outputs in `csv_outputs/` and `screenshots_plots/` for documentation
 - Print concise summaries to the terminal
-

5. Handle Missing/Existing Files Gracefully

- Skip re-saving files if they already exist to avoid overwriting

- Display appropriate warnings if data is missing or inconsistent
-

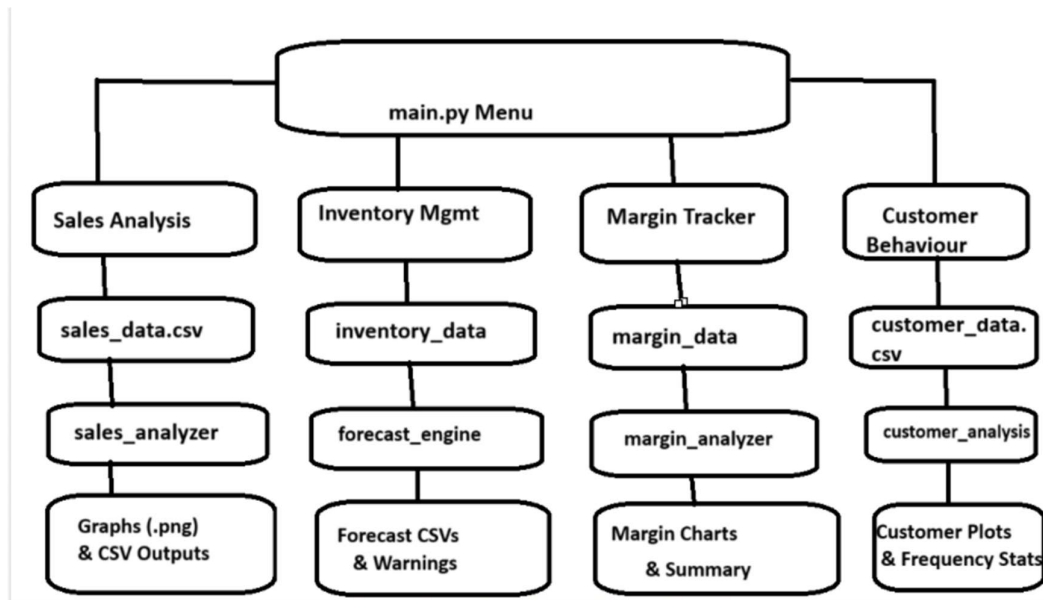
6. Design for Scalability and Reusability

- Use OOP principles inside the models/ folder (e.g., SalesAnalyzer, InventoryManager, etc.)
 - Modular design allows plugging in new datasets or replacing current logic easily
 - CRUD support makes the system interactive and maintainable
-

7. Prepare for GitHub Deployment

- Organize all scripts, outputs, data, and documentation clearly
- Provide requirements.txt
- Create a .zip archive excluding unnecessary files like .pyc, cache, and temp files

1.High Level Architecture Diagram



2. Folder Structure Diagram

King's Coffee Project-2025/

- main.py ← Main menu for analytics and CRUD
- README.md ← Project overview
- requirements.txt ← Python dependencies

kings_coffee_data/ ← Input data (CSV format)

- sales_data.csv
- inventory_data.csv
- customer_data.csv
- margin_data.csv

features/ ← Analysis scripts (menu run)

- sales_analysis.py
- inventory_forecast.py
- margin_optimizer.py
- customer_behaviour.py
- crud_console.py

analysis_moduls/ ← OOP logic and processing classes

- sales_analyzer.py
- forecast_engine.py
- margin_analyzer.py
- customer_analysis.py

crud_operations/ ← CRUD for managing core data

- inventory_crud.py
- sales_crud.py
- customer_crud.py
- margin_crud.py

csv_outputs/ ← Saved analysis CSV summaries

— screenshots_plots/ ← Output graphs and charts

— config/ ← Currently empty (for future configs)

Implementation

This project has been implemented using **modular Python scripts**, **object-oriented programming**, and **data analysis libraries**. Below is a structured overview of how each feature works:

Technologies Used

- **Python 3.11+**
 - **Pandas** – data manipulation
 - **Matplotlib & Seaborn** – data visualization
 - **OS, Sys** – directory handling
 - **OOP** – encapsulation of logic into reusable classes
-

Implementation Modules

1. Sales Analysis

- **Script:** features/sales_analysis.py
- **Class:** analysis_modules/sales_analyzer.py
- **Functions:**
 - Analyze monthly trends
 - Identify top-selling items
 - Summarize sales by cup size
- **Outputs:**
 - Bar plots (sales_trends.png, top_selling_items.png, sales_by_size)
 - CSVs (monthly_sales_summary.csv, top_items.csv)

2. Inventory Forecast

- **Script:** features/inventory_forecast.py
- **Class:** analysis_modules/forecast_engine.py

- **Functions:**

- Forecast stock requirements
- Calculate average usage per item

- **Outputs:**

- inventory_forecast.csv
- Plot: (inventory_forecast.png, weekly_inventory_forecast.png)

3. Margin Optimization

- **Script:** features/margin_optimizer.py

- **Class:** analysis_modules/margin_analyzer.py

- **Functions:**

- Analyze cost vs price per item
- Optimize selling price to increase profit margin

- **Outputs:**

- Margin_report.csv
- Plots: low_margin_items.png

4. Customer Behavior

- **Script:** features/customer_behaviour.py

- **Class:** analysis_modules/customer_analysis.py

- **Functions:**

- Identify top spenders
- Analyze visit frequency
- Detect loyal customers

- **Outputs:**

- Graph: monthly_visits.png
- CSVs: returning_customers.csv, top_spenders.csv

5. CRUD Operations

- **Scripts:**
 - crud_console.py (menu)
 - crud_operations/ folder (4 files)
 - **Features:**
 - Add, update, delete, view entries for:
 - Inventory
 - Sales
 - Customers
 - Margins
 - **Persistence:** Edits saved directly to the respective CSV files
-

Main Interface

- File: main.py
- Acts as the **entry point** for the full system
- Allows users to select:
 1. Sales Analysis
 2. Inventory Forecast
 3. Margin Optimization
 4. Customer Behavior Analysis
 5. CRUD Console
 6. Exit

Code and Explanation

1. Sales Analysis

File: features/sales_analysis.py

Class: SalesAnalyzer in analysis_modules/sales_analyzer.py

Sample Snippet:

python

Copy code

```
monthly_sales =  
self.data.groupby(self.data['date'].dt.strftime('%b'))['total_price'].sum()
```

Explanation:

- Groups the data by month abbreviation (e.g., Jan, Feb)
 - Aggregates the total sales (total_price) for each month
-

2. Inventory Forecast

File: features/inventory_forecast.py

Class: ForecastEngine in analysis_modules/forecast_engine.py

Sample Snippet:

python

Copy code

```
average_usage = self.data.groupby("item")["quantity"].mean()  
forecast = average_usage * forecast_months
```

Explanation:

- Calculates the average usage of each item
 - Forecasts inventory needed by multiplying with future months
-

3. Margin Optimization

File: features/margin_optimizer.py

Class: MarginAnalyzer in analysis_modules/margin_analyzer.py

Sample Snippet:

python

Copy code

```
df["profit_margin"] = df["price"] - df["cost"]  
df["suggested_price"] = df["cost"] + df["profit_margin"].mean()
```

Explanation:

- Calculates profit margin per item
 - Suggests a uniform pricing strategy by applying average margin
-

4. Customer Behavior Analysis

File: features/customer_behaviour.py

Class: CustomerBehavior in analysis_modules/customer_analysis.py

Sample Snippet:

python

Copy code

```
visit_data["month"] =  
pd.to_datetime(visit_data["visit_date"]).dt.month_name()  
visit_freq = visit_data["month"].value_counts().sort_index()
```

Explanation:

- Extracts the month from visit_date
 - Counts number of customer visits per month
-

5. CRUD Example – Customer

File: crud_operations/customer_crud.py

Sample Snippet:

python

Copy code

```
df = pd.read_csv(data_path)
df = df[["customer_id", "visit_date", "amount"]]
```

Explanation:

- Loads customer data
 - Keeps only the required columns to ensure consistency and prevent NaNs
-

6. Main Menu

File: main.py

Sample Snippet:

python

Copy code

```
choice = input("Select an option (1–6): ")
if choice == "1":
    from features import sales_analysis; sales_analysis.run()
```

Explanation:

- Takes user input to decide which analysis to run
 - Dynamically calls the appropriate run() function
-

The modular structure ensures separation of concerns:

- **Features/** handles user-facing logic
- **analysis_modules/** holds reusable backend logic
- **crud_operations/** manages direct data file modifications
- **main.py** links everything through a user menu

Output Screenshots

1.Sales Analysis Output

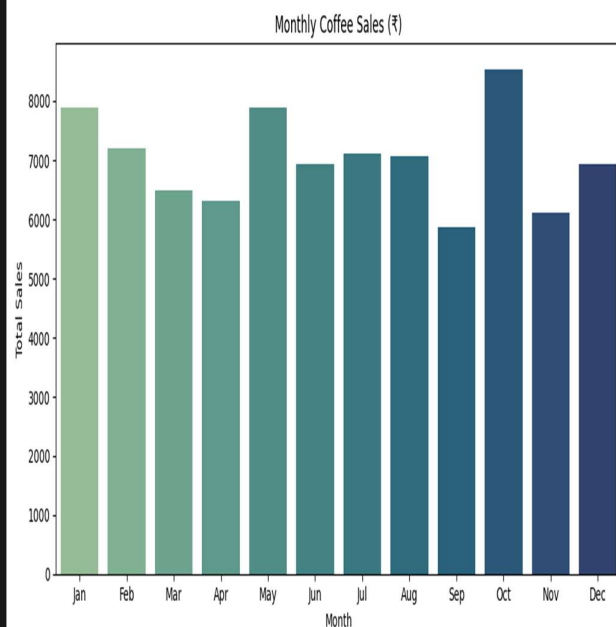
Monthly Sales Summary

```
==== KING'S COFFEE SHOP ANALYTICS MENU ====
1. Sales Analysis
2. Inventory Forecast
3. Margin Optimization
4. Customer Behavior Analysis
5. CRUD Console
6. Exit
Select an option (1-6): 1
```

Running: Sales Analysis

MONTHLY SALES SUMMARY

month	
Jan	7880.0
Feb	7200.0
Mar	6500.0
Apr	6570.0
May	7880.0
Jun	6940.0
Jul	7120.0
Aug	7060.0
Sep	5880.0
Oct	8540.0
Nov	6120.0
Dec	6940.0

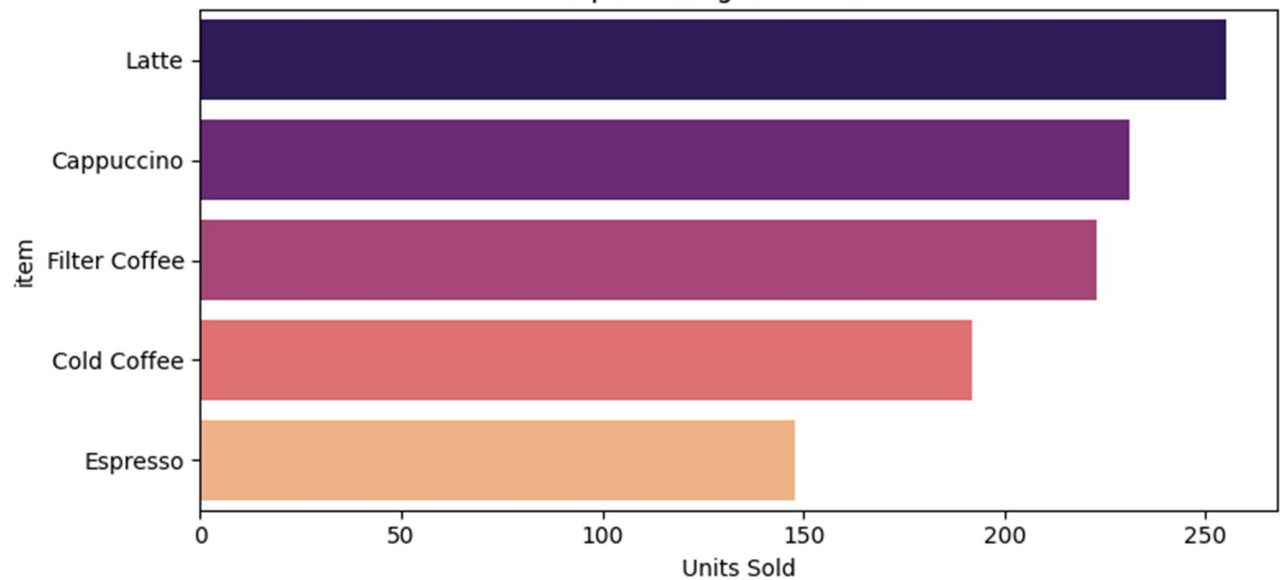


Top 5 Selling Items

TOP 5 SELLING ITEMS

item	
Latte	255
Cappuccino	231
Filter Coffee	223
Cold Coffee	192
Espresso	148

Top 5 Selling Coffee Items

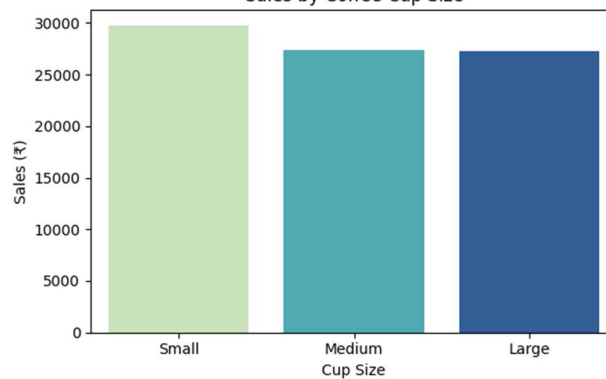


Sales by Cup Size

SALES BY CUP SIZE

size	
Small	29740.0
Large	27490.0
Medium	27400.0

Sales by Coffee Cup Size

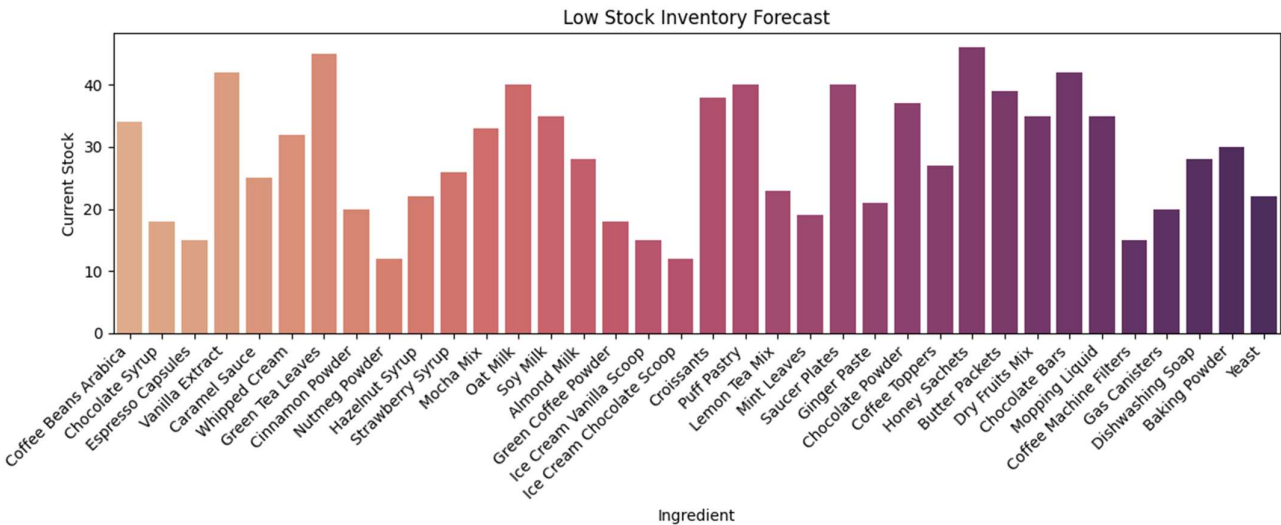


2.Inventory Forecast Output

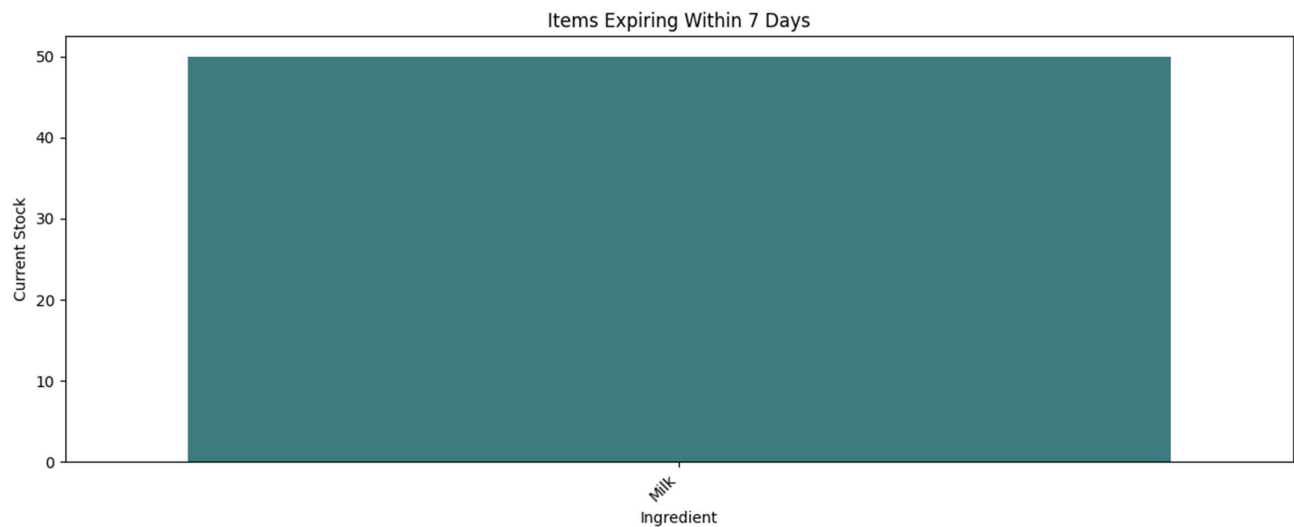
Sample Forecast

Running: Inventory Forecast

Inventory Forecast											
	Ingredient	CostPrice	SellingPrice	expiry_date	restock_date	CurrentStock	item	quantity	daily_usage	Status	
0	Coffee Beans Arabica	120.0	200.0	2025-08-10	2025-07-10	34.0	NaN	NaN	NaN	Low Stock	
3	Chocolate Syrup	40.0	70.0	2025-08-12	2025-07-12	18.0	NaN	NaN	NaN	Low Stock	
5	Espresso Capsules	35.0	60.0	2025-07-30	2025-07-16	15.0	NaN	NaN	NaN	Low Stock	
6	Vanilla Extract	90.0	140.0	2025-08-15	2025-07-13	42.0	NaN	NaN	NaN	Low Stock	
7	Caramel Sauce	55.0	90.0	2025-08-20	2025-07-16	25.0	NaN	NaN	NaN	Low Stock	
8	Whipped Cream	20.0	35.0	2025-07-26	2025-07-12	32.0	NaN	NaN	NaN	Low Stock	
10	Green Tea Leaves	40.0	60.0	2025-08-12	2025-07-11	45.0	NaN	NaN	NaN	Low Stock	
15	Cinnamon Powder	15.0	25.0	2025-08-18	2025-07-10	20.0	NaN	NaN	NaN	Low Stock	
16	Nutmeg Powder	18.0	30.0	2025-08-22	2025-07-12	12.0	NaN	NaN	NaN	Low Stock	
17	Hazelnut Syrup	50.0	85.0	2025-08-25	2025-07-14	22.0	NaN	NaN	NaN	Low Stock	
18	Strawberry Syrup	45.0	75.0	2025-08-28	2025-07-13	26.0	NaN	NaN	NaN	Low Stock	
19	Mocha Mix	65.0	100.0	2025-08-27	2025-07-15	33.0	NaN	NaN	NaN	Low Stock	
20	Oat Milk	30.0	50.0	2025-07-28	2025-07-16	40.0	NaN	NaN	NaN	Low Stock	
21	Soy Milk	28.0	48.0	2025-07-29	2025-07-16	35.0	NaN	NaN	NaN	Low Stock	
22	Almond Milk	32.0	55.0	2025-08-01	2025-07-15	28.0	NaN	NaN	NaN	Low Stock	
23	Green Coffee Powder	60.0	95.0	2025-08-18	2025-07-14	18.0	NaN	NaN	NaN	Low Stock	
26	Ice Cream Vanilla Scoop	25.0	45.0	2025-08-05	2025-07-11	15.0	NaN	NaN	NaN	Low Stock	
27	Ice Cream Chocolate Scoop	25.0	45.0	2025-08-06	2025-07-11	12.0	NaN	NaN	NaN	Low Stock	
33	Croissants	20.0	35.0	2025-08-17	2025-07-15	38.0	NaN	NaN	NaN	Low Stock	
34	Puff Pastry	10.0	18.0	2025-08-18	2025-07-15	40.0	NaN	NaN	NaN	Low Stock	
35	Lemon Tea Mix	35.0	60.0	2025-08-20	2025-07-15	23.0	NaN	NaN	NaN	Low Stock	
36	Mint Leaves	5.0	10.0	2025-07-25	2025-07-16	19.0	NaN	NaN	NaN	Low Stock	
39	Saucer Plates	10.0	20.0	2026-01-01	2025-07-11	40.0	NaN	NaN	NaN	Low Stock	
44	Ginger Paste	15.0	25.0	2025-07-25	2025-07-13	21.0	NaN	NaN	NaN	Low Stock	
46	Chocolate Powder	20.0	35.0	2025-08-05	2025-07-11	37.0	NaN	NaN	NaN	Low Stock	
47	Coffee Toppers	6.0	10.0	2025-08-15	2025-07-12	27.0	NaN	NaN	NaN	Low Stock	
48	Honey Sachets	12.0	18.0	2025-08-17	2025-07-12	46.0	NaN	NaN	NaN	Low Stock	
49	Butter Packets	10.0	15.0	2025-08-18	2025-07-12	39.0	NaN	NaN	NaN	Low Stock	
51	Dry Fruits Mix	20.0	30.0	2025-08-21	2025-07-13	35.0	NaN	NaN	NaN	Low Stock	
53	Chocolate Bars	15.0	25.0	2025-08-26	2025-07-13	42.0	NaN	NaN	NaN	Low Stock	
57	Mopping Liquid	25.0	40.0	2025-11-01	2025-07-11	35.0	NaN	NaN	NaN	Low Stock	
60	Coffee Machine Filters	12.0	18.0	2025-12-01	2025-07-12	15.0	NaN	NaN	NaN	Low Stock	
61	Gas Canisters	150.0	180.0	2025-11-15	2025-07-12	20.0	NaN	NaN	NaN	Low Stock	
62	Dishwashing Soap	45.0	60.0	2025-10-10	2025-07-12	28.0	NaN	NaN	NaN	Low Stock	
64	Baking Powder	10.0	15.0	2025-09-05	2025-07-12	30.0	NaN	NaN	NaN	Low Stock	
65	Yeast	8.0	12.0	2025-09-06	2025-07-12	22.0	NaN	NaN	NaN	Low Stock	



Inventory Expiring Within 7 days



3.Margin Optimization Output

Margin optimization Report

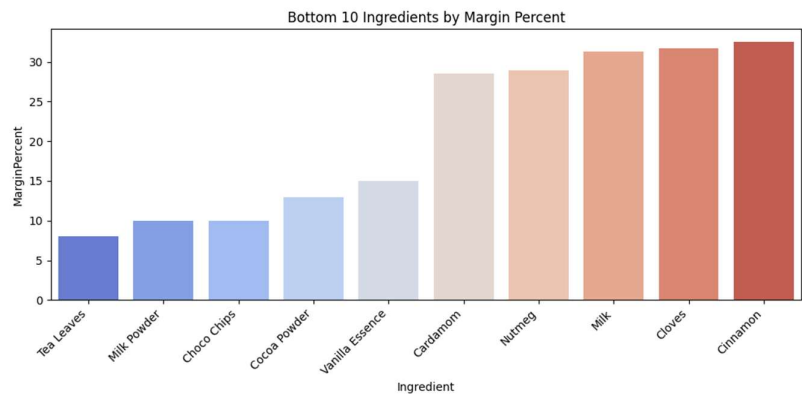
```
=== KING'S COFFEE SHOP ANALYTICS MENU ===
1. Sales Analysis
2. Inventory Forecast
3. Margin Optimization
4. Customer Behavior Analysis
5. CRUD Console
6. Exit
Select an option (1-6): 3

Running: Margin Optimization

MARGIN OPTIMIZATION REPORT
```

	Ingredient	supplier	CostPrice	SellingPrice	last_updated	item	cost_price	selling_price	Profit	MarginPercent
29	Napkins	FreshTouch	0.5	2.0	2025-06-18	NaN	NaN	NaN	1.5	75.000000
27	Straws	StrawLine	1.0	3.0	2025-06-15	NaN	NaN	NaN	2.0	66.666667
60	Soda	FreshFizz	15.0	40.0	2025-06-18	NaN	NaN	NaN	25.0	62.500000
76	Chutney	ChilliChatni	15.0	40.0	2025-06-21	NaN	NaN	NaN	25.0	62.500000
63	Water Bottle	Aquapure	8.0	20.0	2025-06-19	NaN	NaN	NaN	12.0	60.000000
..
88	Vanilla Essence	EssenceMart	255.0	300.0	2025-06-30	NaN	NaN	NaN	45.0	15.000000
89	Cocoa Powder	ChocoFarm	270.0	310.0	2025-07-03	NaN	NaN	NaN	40.0	12.903226
85	Milk Powder	DairyBest	180.0	200.0	2025-07-01	NaN	NaN	NaN	20.0	10.000000
86	Choco Chips	DelightFoods	135.0	150.0	2025-06-28	NaN	NaN	NaN	15.0	10.000000
87	Tea Leaves	GreenGold	230.0	250.0	2025-07-05	NaN	NaN	NaN	20.0	8.000000

Low Margin Ingredients



4.Customer Behavior Output

Running: Customer Behavior Analysis

Running: Customer Behavior Analysis

Returning Customers:

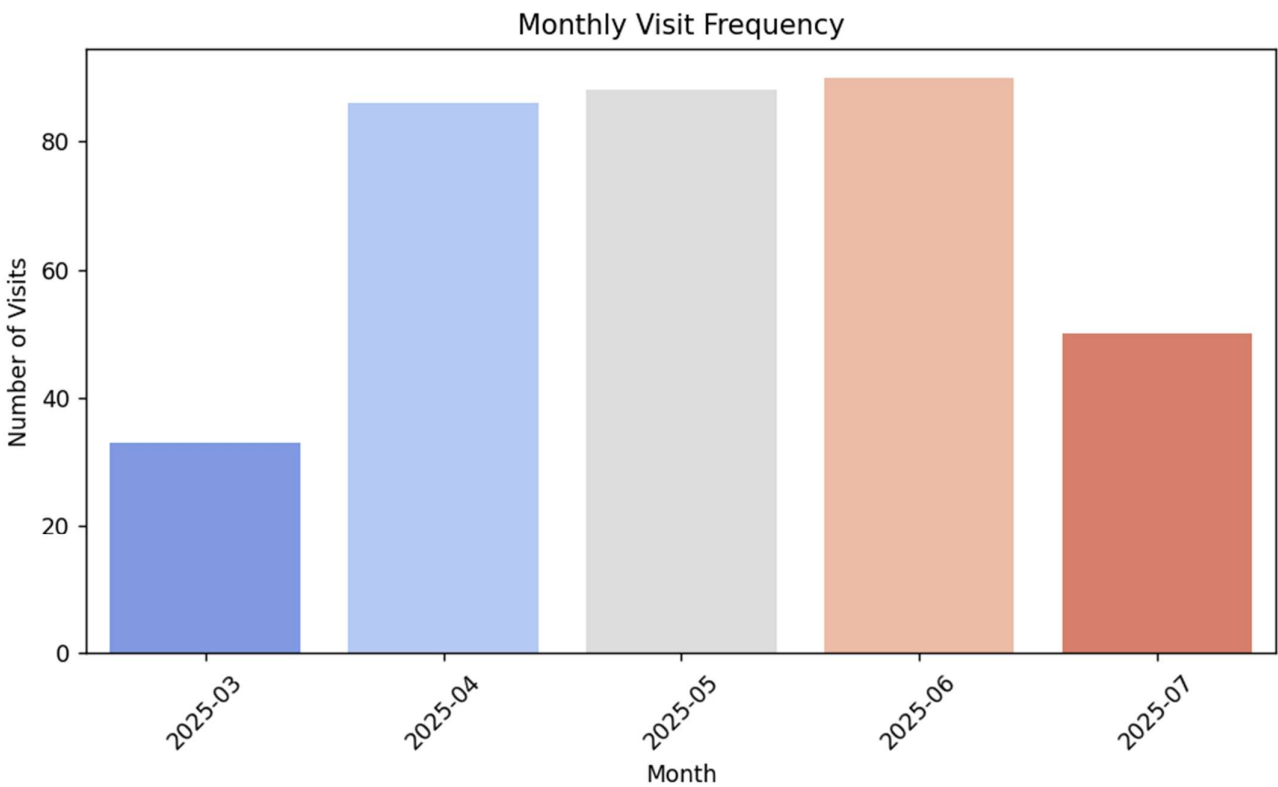
	customer_id	visit_date	amount
307	CUST0021	2025-03-19	116.66
39	CUST0022	2025-03-19	371.46
25	CUST0009	2025-03-20	186.57
172	CUST0044	2025-03-20	255.48
66	CUST0044	2025-03-21	186.33

Top 10 Spenders:

	customer_id	amount
0	CUST0024	3557.88
1	CUST0044	3036.30
2	CUST0031	2704.01
3	CUST0048	2679.12
4	CUST0040	2666.88
5	CUST0018	2629.59
6	CUST0030	2546.17
7	CUST0010	2490.81
8	CUST0042	2490.35
9	CUST0041	2450.97

Monthly Visit Frequency:

Month	
2025-03	33
2025-04	86
2025-05	88
2025-06	90
2025-07	50



Closure

King's Coffee Shop Analytics is a comprehensive, menu-driven data analytics system that enables actionable insights for improving operations in a mid-sized Indian coffee shop. The project uses real-world-style datasets and applies the four pillars of data analytics—Descriptive, Diagnostic, Predictive, and Prescriptive Analytics—to deliver impactful business decisions.

By implementing this project, the shop owner or manager can:

- Identify high-selling items and peak seasons.**
- Forecast inventory to prevent overstocking and understocking.**
- Optimize pricing and margin strategies based on profitability.**
- Understand customer loyalty and purchasing patterns to offer personalized promotions.**

With an integrated CRUD console, the project also allows real-time management of inventory, sales, customer data, and margins, ensuring that the system is dynamic and future-ready.

This project serves not just as a solution for one coffee shop, but as a blueprint for similar data-driven businesses aiming to modernize their decision-making using Python, Pandas, Matplotlib, and Seaborn—all within a modular and well-documented structure.

Bibliography

1. Pandas Documentation – <https://pandas.pydata.org/>
2. Matplotlib Documentation – <https://matplotlib.org/stable/contents.html>
3. Seaborn Documentation – <https://seaborn.pydata.org/>
4. Python Official Docs – <https://docs.python.org/3/>
5. Coffee industry pricing data – Mock data inspired by Indian cafés (e.g., Starbucks India, Third Wave Coffee, Chai Point)

*****End of Project*****
