PROJECT: KANTI SWEETS DATA ANALYSIS

K Faheem Ahmed

PRESIDENCY UNIVERSITY

1. Problem Statement:

Optimizing Inventory and Maximizing Revenue for Kanti Sweets

Kanti Sweets, a well-established sweet and savoury manufacturer with over 135 outlets, faces the critical challenge of optimizing its perishable inventory and maximizing revenue. Given the short shelf life of many traditional Indian sweets and the seasonal fluctuations in demand driven by festivals, inefficient inventory management can lead to significant waste and missed sales opportunities. Currently, Kanti Sweets lacks a systematic approach to leverage its sales data for strategic decision-making, leading to potential overproduction of less popular items, stockouts of high-demand sweets, and suboptimal revenue generation throughout the year.

The core problem is the absence of comprehensive data analysis that can provide actionable insights into product performance, seasonal sales patterns, and inventory efficiency, thereby hindering Kanti Sweets ability to minimize wastage and maximize profitability across its diverse product range.

2. Detailed Description of the Project:

This project aims to implement a data-driven approach to analyze Kanti Sweets' sales and inventory data to address the aforementioned problem. The analysis will focus on providing clear, actionable insights into product performance, revenue trends, and waste reduction strategies.

Objective 1: Identify Best-Selling Sweets in Specific Months

- **Description:** This objective involves analyzing historical sales data to determine which specific sweets experience the highest sales volume during particular months. This analysis will go beyond overall annual sales to uncover seasonal preferences.
- **Data Required:**
    - sweet_id: Unique identifier for each sweet.
    - sale_date: Date of sale (including month and year).
    - quantity_sold: Number of units or weight of the sweet sold.
    - unit_price: Price per unit/weight of the sweet.
- **Expected Output:** Monthly reports or dashboards indicating the top N selling sweets for each month, This will help Kanti Sweets align production and marketing efforts with consumer demand during different seasons and festivals. For instance, knowing

that "Motichoor Laddu" sells exceptionally well in October due to Diwali will enable optimized production for that period.

Objective 2: Determine Periods of Highest Company Revenue:

- **Description:** This objective focuses on identifying the specific months or quarters when Kanti Sweets generates the highest revenue. Understanding these peak periods is crucial for strategic planning, resource allocation, and targeted marketing campaigns.
- **Data Required:**
    - sale_date: Date of sale.
    - total_transaction_value: Total revenue generated per transaction or aggregated daily/monthly revenue.
    - (Derived from quantity_sold * unit_price per sweet_id and summing across all sweets for a given period).
- **Expected Output:** A clear visualization (e.g., line graph, bar chart) showing monthly/quarterly revenue trends over multiple years. This will highlight revenue peaks (e.g., during major festivals like Diwali, Dussehra, Sankranti, or special occasions) and troughs, allowing Kanti Sweets to strategically increase staff, stock, and promotional activities during high-revenue periods and plan for leaner times.

Objective 3: Conduct Yearly Sales Data Comparison

- **Description:** This objective involves comparing total sales performance year-on-year. This long-term view will help assess overall business growth, identify macro trends, and evaluate the effectiveness of past business strategies.
- **Data Required:**
    - sale_date: Date of sale.
    - total_transaction_value or aggregated total_sales_value per year.
- **Expected Output:** A comparative analysis of total annual sales figures (e.g., bar charts, growth percentages) for available historical years. This will demonstrate whether the company is growing, stagnating, or declining, providing a high-level overview of business health.

Objective 4: Identify Sweets with the Most Wastage (Sold Less)

- **Description:** This crucial objective aims to pinpoint which sweets are frequently overproduced or stocked in excess, leading to spoilage, reduced freshness, and financial loss. This requires combining sales data with production or inventory data.
- **Data Required:**
    - sweet_id: Unique identifier for each sweet.

    - quantity_produced_or_stocked: Amount of sweet produced or received into inventory for a given period.
    - quantity_sold: Amount of sweet sold in the same period.

- **Expected Output:** A list or ranking of sweets with the highest discrepancy between production/stock and sales, indicating potential overproduction and waste. This might be presented as a "waste percentage" for each sweet. This insight will enable Kanti Sweets to adjust production schedules, optimize procurement of raw materials, introduce dynamic pricing for slow-moving items, or even discontinue consistently low-performing or high-waste products.

Overall Impact:

By successfully addressing these objectives, Kanti Sweets can move from reactive to proactive inventory and sales management. This will lead to:

- **Reduced Wastage:** Minimizing losses from unsold and expired perishable sweets.
- **Increased Profitability:** Optimizing production to meet demand, leading to higher sales of popular items and efficient resource utilization.
- **Improved Customer Satisfaction:** Ensuring availability of popular sweets and freshness of products.
- **Enhanced Strategic Planning:** Providing data-driven insights for marketing, promotions, and overall business development, particularly around key festive seasons.
- **Better Resource Allocation:** Optimizing labour, raw material procurement, and storage based on accurate demand forecasts.

3. Design:

kanti_sweets_analysis/

│

├── data/

│ └── kanti_sweets_dummy_sales_data.csv

│

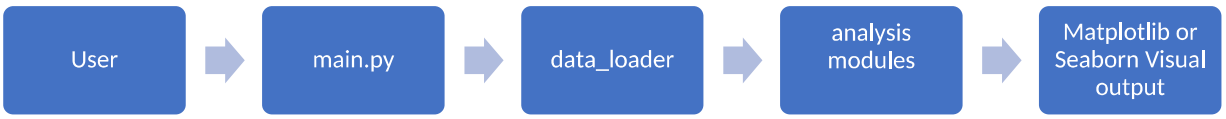
├── utils/

│ └── data_loader.py

│

```
├── analysis/
│   ├── monthly_comparison.py
│   ├── revenue_analysis.py
│   ├── top_selling_sweet.py
│   ├── waste_analysis.py
│
│
├── main.py
└── requirements.txt
```

4.Flow diagram:



5.Code and implementation:

```python
import matplotlib.pyplot as plt

def sweet_sales_pm(df, month):
    df_month = df[df['month'] == month]
    sweet_totals = df_month.groupby('sweet_name')['quantity_sold'].sum()
    sweet_totals.plot.pie(autopct='%1.1f%%', startangle=90, figsize=(7,7))
    plt.title(f"Top Selling Sweets in Month {month}")
    plt.ylabel("")
    plt.tight_layout()
    plt.show()
```

The sweet_sales_pm function is designed to visually represent the sales distribution of sweets in a specific month using a pie chart. It helps identify which sweets sold the most or least in that selected time frame.

```python
import seaborn as sns
import matplotlib.pyplot as plt

def monthly_sales_bar(df):
    monthly_sales = df.groupby(['month'])['quantity_sold'].sum()
    sns.barplot(x=monthly_sales.index, y=monthly_sales.values, palette='muted')
    plt.title("Monthly Total Sweet Sales")
    plt.xlabel("Month")
    plt.ylabel("Quantity Sold")
    plt.tight_layout()

    plt.show()

def monthly_sales_by_sweet(df):
    pivot = df.pivot_table(values='quantity_sold', index='month', columns='sweet_name', aggfunc='sum').fil
    pivot.plot(kind='bar', stacked=True, figsize=(12, 6))
    plt.title("Monthly Sweet Sales Comparison")
    plt.xlabel("Month")
    plt.ylabel("Quantity Sold")
    plt.tight_layout()
    plt.show()
```

Function Name	Visualization Type	Focus	Benefit
monthly_sales_bar()	Bar Chart (total)	Overall monthly sales	Identify peak and low sales months
monthly_sales_by_sweet()	Stacked Bar Chart	Sweet-wise monthly breakdown	Compare sweets across months

```python
def highest_revenue_month(df):
    monthly_revenue = df.groupby('month')['revenue'].sum()
    best_month = monthly_revenue.idxmax()
    best_value = monthly_revenue.max()
    print(f"📈 Highest Revenue Month: {best_month} with ₹{best_value}")
```

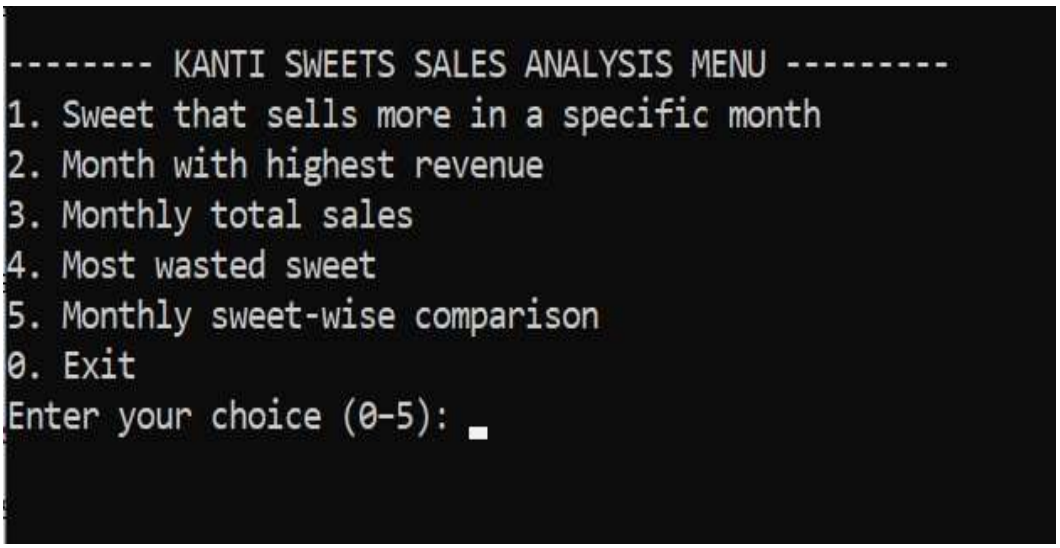
The highest_revenue_month(df) function identifies the month with the highest total revenue from sweet sales. It groups the dataset by month and calculates the sum of revenue for each month. Using this grouped data, it determines which month had the maximum revenue (idxmax()) and prints both the month and the corresponding revenue value. This function helps the business quickly pinpoint the most profitable period of the year, which is valuable for financial planning, marketing focus, and resource allocation.

```python
import matplotlib.pyplot as plt
def most_wasted_sweet(df):
    total_sales = df.groupby('sweet_name')['quantity_sold'].sum().sort_values()
    total_sales.plot(kind='barh', color='tomato', figsize=(10, 6))
    plt.title("Least to Most Sold Sweets in the whole year ")
    plt.xlabel("Total Quantity Sold")
    plt.ylabel("Sweet Name")
    plt.tight_layout()
    plt.show()
```

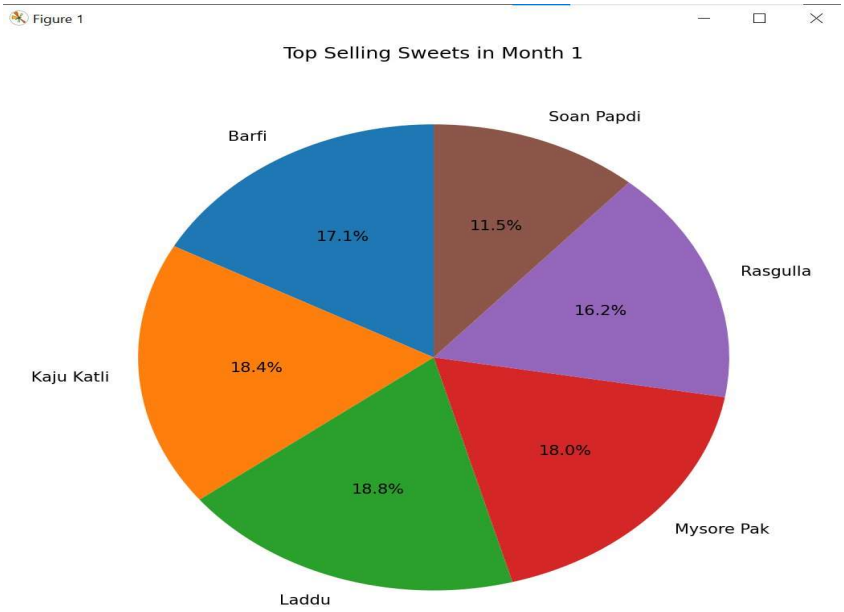
The most_wasted_sweet(df) function visualizes which sweets were sold the least over the entire year, helping to identify the most wasted or unpopular items. It groups the data by sweet name, sums up the total quantity sold for each, and sorts them in ascending order. The results are displayed as a horizontal bar chart, with sweets selling the least appearing at the top. This analysis is crucial for inventory optimization, helping the business reduce waste and focus on more popular products.

6. Output screenshots:

1.Menu



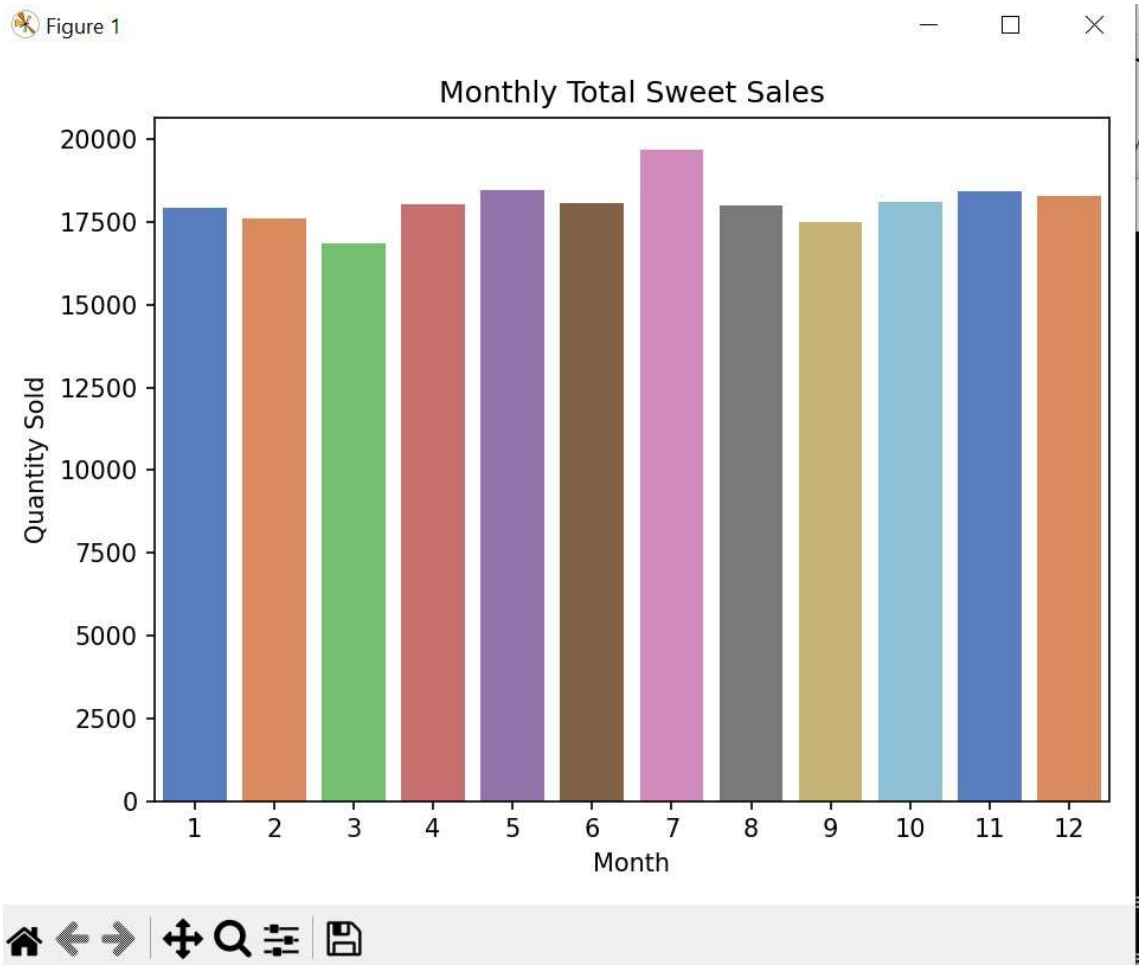
2.Top selling sweet for specified month:



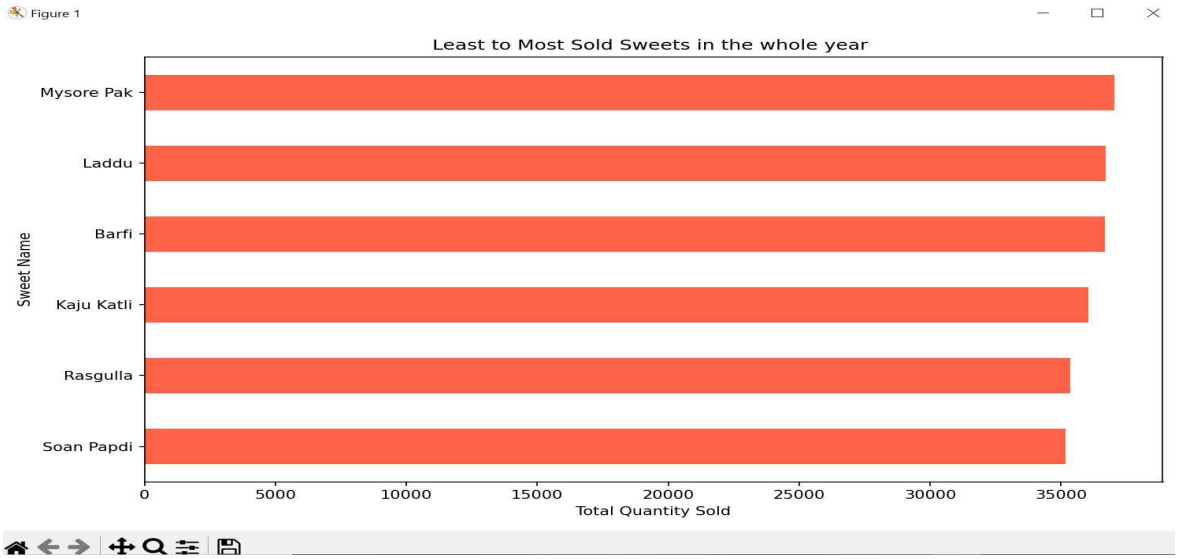
3.Month with highest revenue:

```
-------- KANTI SWEETS SALES ANALYSIS MENU ---------
1. Sweet that sells more in a specific month
2. Month with highest revenue
3. Monthly total sales
4. Most wasted sweet
5. Monthly sweet-wise comparison
0. Exit
Enter your choice (0-5): 2
  Highest Revenue Month: 7 with ₹667789
```

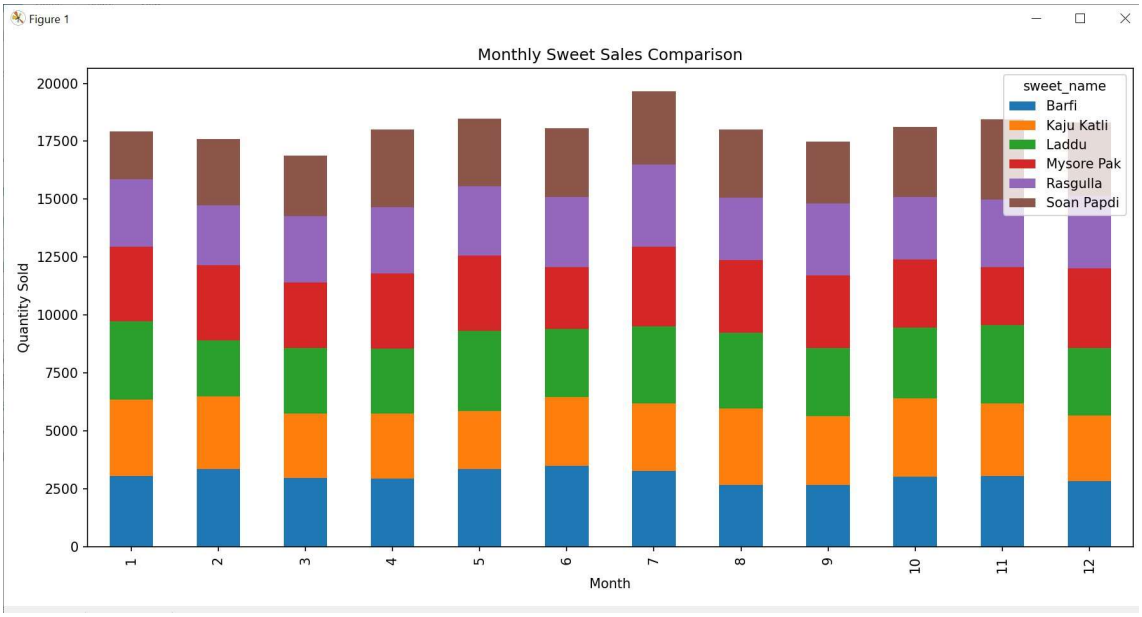
4.Monthly total sales:



5. Least sold sweet:



6.Monthly sales comparision:



7. Conclusion:

In conclusion, this project provides a robust framework for Kanti Sweets to transition from traditional inventory management to a data-driven operational strategy. By meticulously analyzing sales and inventory data, Kanti Sweets will gain unparalleled clarity on product performance, seasonal demand fluctuations, and areas of significant wastage. The insights derived from identifying best-selling sweets by month, pinpointing peak revenue periods, conducting yearly sales comparisons, and detecting high-wastage items will empower the company to make precise adjustments to production, procurement, and marketing efforts. Ultimately, this will lead to a substantial reduction in perishable sweet wastage, optimized inventory levels, increased profitability, and a more responsive and efficient supply chain across all 135+ outlets. This analytical approach is critical for Kanti Sweets to sustain growth and maintain its market leadership in the dynamic Indian sweet industry.

Bibliography:

1. McKinney, Wes. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media, 2018.
   – Used for understanding data manipulation and analysis with pandas.
2. Hunter, J. D. *Matplotlib: A 2D Graphics Environment*. Computing in Science & Engineering, 2007.
   – Reference for plotting graphs and charts using matplotlib.
3. Waskom, M. L. *Seaborn: Statistical Data Visualization*. Journal of Open Source Software, 2021.
   – Used for creating advanced visualizations like bar plots and line plots using seaborn.
4. Python Software Foundation. *Python Language Reference, version 3.10*. https://www.python.org
   – For syntax and language features used throughout the program.
5. Stack Overflow (https://stackoverflow.com)
   – For resolving coding errors and learning best practices during development.
6. Official Pandas Documentation. https://pandas.pydata.org/docs
   – For methods used in data filtering, grouping, and pivoting.
7. Matplotlib Documentation. https://matplotlib.org/stable/contents.html
   – For customizing and improving the appearance of plots.
8. Seaborn Documentation. https://seaborn.pydata.org
   – For styling and structuring visual outputs related to monthly sales.