Coffee Shop Sales Analysis Project

Name - Abhay Kumar | Email - meabhaykr@gmail.com

Project Overview

The **Coffee Shop Sales Analysis Project** replicates an Excel dashboard in Python to provide insights into sales, visitor behaviour, and product performance. This project uses Python libraries like pandas, matplotlib, and seaborn for data analysis and visualization. The project is designed to analyze sales data and present actionable insights through interactive charts and metrics.

Colab Link:
https://colab.research.google.com/drive/1ZmcHhHAD8Dq_A7J5QI9QXkL_f_Ytts8i?usp=sharing

---

Objective

1. Analyze sales data to uncover trends in visitor behaviour and product performance.
2. Visualize the data for better understanding using Python tools.
3. Replicate the Excel dashboard functionality in a Python environment.

Dataset

Expected Dataset Structure:

The dataset should contain the following columns:

- transaction_id: Unique identifier for each transaction.
- transaction_date: Date of the transaction.
- transaction_time: Time of the transaction.
- store_id: Unique identifier for each store.
- store_location: Store location name.
- product_id: Unique identifier for each product.
- transaction_qty: Quantity of products sold.
- unit_price: Price per unit of the product.
- product_category: Category of the product (e.g., Coffee, Bakery).
- product_type: Type of product (e.g., Beverage, Food).
- product_detail: Product name or description.
- size: Size of the product (e.g., Small, Medium, Large).

- amount: Total amount for the transaction.
- month_name: Month of the transaction.
- day_name: Day of the transaction.
- hour: Hour of the transaction.
- day_of_week: Numeric day of the week (e.g., 1 = Monday).
- month: Numeric month (e.g., 1 = January).

Features

1. Metrics Calculation

- **Total Sales**: Sum of all transaction amounts.
- **Total Visitors**: Number of unique transaction IDs.
- **Average Bill**: Average amount per transaction.
- **Average Orders**: Average quantity of products per transaction.

2. Visualizations

a. Quantity Ordered by Hours

- A line chart showing the quantity of products ordered across different hours of the day.

b. Footfall Over Store Locations

- A bar chart displaying the transaction count and total sales for each store location.

c. Size Distribution Based on Orders

- A pie chart showcasing the percentage distribution of product sizes (e.g., Small, Regular, Large).

d. Category % Distribution Based on Sales

- A pie chart illustrating the percentage contribution of each product category to the total sales.

e. Numbers of Sales Based on Weekdays

- A bar chart comparing the quantity of products sold on different days of the week.

f. Top 5 Products Based on Sales

- A horizontal bar chart highlighting the top 5 products by sales amount.

Technical Requirements

Libraries Used:

- **pandas**: For data manipulation and aggregation.

- **matplotlib**: For creating visualizations.
- **seaborn**: For enhancing chart aesthetics.

Installation:

Run the following command to install the required libraries:

pip install pandas matplotlib seaborn openpyxl

Python Code

Refer to the Python script provided earlier for the full implementation.

How to Use

1. **Prepare Your Dataset**: Ensure your data is saved in an Excel or CSV file with the required columns.
2. **Set Up the Environment**:
    - Install Python and necessary libraries.
    - Save the dataset in the project folder.
3. **Run the Script**:
    - Replace "data_file.xlsx" in the script with your dataset's path.
    - Execute the script in a Python IDE or Jupyter Notebook.
4. **Visualize the Results**:
    - The script generates interactive charts and metrics for analysis.

Output Examples

1. Metrics

- **Total Sales**: ₹698,800
- **Total Visitors**: 149,120
- **Average Bill**: ₹4.69
- **Average Orders**: 1.44

2. Charts

- **Line Chart**: Quantity Ordered by Hours.
- **Bar Chart**: Footfall Over Store Locations.
- **Pie Chart**: Size and Category Distribution.

- **Horizontal Bar Chart**: Top 5 Products by Sales.

Future Enhancements

1. Add interactivity using tools like **Dash** or **Plotly**.
2. Implement filtering by store, date range, or product category.
3. Automate report generation in PDF or HTML format.