Data Analytics on Car Wash Business Model

Index

Description

Data & Problem Context

Plan

Design

Implementation

Code with Explanation

Output Screenshots

Closure

1. Description

Project Title:

Customer & Sales Analysis of a Car Wash Business

This project delivers data-driven insights into a car wash business using transaction records. It analyzes sales, customer retention, seasonal demand, and service performance while also visualizing time-of-day trends and carbon footprint.

2. Data & Problem Context

**Problem Statement**

How can we maximize revenue and retain valuable customers through data analytics?

Objectives:

Customer churn rate

Service package recommendation

Seasonality trends

Profitability by service

Identify low-sales months

Carbon footprint tracker

Rank the most-used and highest-earning services:

3. Plan

* Load the data
* Preprocess timestamp info
* Analyze churn, seasonality, and low-revenue months
* Visualize findings via pie/bar/line charts
* Use modular architecture for maintainability

4. Design

**Modular Components:**

* CarWashApp: Controls workflow
* CarWashAnalyzer: Performs calculations
* CarWashVisualizer: Generates visual insights

**Data Flow Diagram:**

User

↓

CarWashApp Menu

↓

Preprocessing → CarWashAnalyzer → Insights

↓

CarWashVisualizer → Charts

5. Implementation

**Technologies Used:**

* Language: Python
* Libraries: pandas, matplotlib
* Architecture: OOP
* Data Source: Local CSV

6. Code with Explanation

Main Application (CarWashApp.py)

import pandas as pd

from analysis import CarWashAnalyzer

from visualization import CarWashVisualizer

# Dataset path

DATA\_PATH = r"C:\LEARNING\sic\_pu\_june25\hackathon\car\_wash data set\car\_wash\_final\_dataset\_with\_date (1).csv"

class CarWashApp:

def \_\_init\_\_(self):

self.df = None

self.analyzer = None

self.visualizer = CarWashVisualizer()

def preprocess\_data(self):

print("\nPreprocessing data...")

self.df = pd.read\_csv(DATA\_PATH, parse\_dates=["Transaction Date & Time"])

self.df['Month'] = self.df['Transaction Date & Time'].dt.month\_name()

self.df['DayOfWeek'] = self.df['Transaction Date & Time'].dt.day\_name()

self.df['Hour'] = self.df['Transaction Date & Time'].dt.hour

self.df['Time Slot'] = self.df['Hour'].apply(

lambda h: "Night" if h < 6 else "Morning" if h < 12 else "Afternoon" if h < 18 else "Evening"

)

self.df.to\_csv(DATA\_PATH, index=False)

self.analyzer = CarWashAnalyzer(self.df)

print("Preprocessing complete.")

def show\_menu(self):

while True:

print("\n--- Car Wash Analysis Menu ---")

print("1. Analyze Customer Churn")

print("2. Recommend Service Packages")

print("3. Analyze Seasonality")

print("4. Track Carbon Footprint")

print("5. Identify Low-Sales Month")

print("6. Rank Services")

print("7. Show All Visualizations")

print("0. Exit")

choice = input("Enter your choice: ")

if choice == "1":

self.analyzer.analyze\_churn\_rate()

elif choice == "2":

self.analyzer.recommend\_service\_packages()

elif choice == "3":

self.analyzer.analyze\_seasonality()

elif choice == "4":

self.analyzer.track\_carbon\_footprint()

elif choice == "5":

self.analyzer.identify\_low\_sales\_month()

elif choice == "6":

self.analyzer.rank\_services()

elif choice == "7":

self.visualizer.show\_all\_plots()

elif choice == "0":

print("Exiting...")

break

else:

print("Invalid choice. Please try again.")

if \_\_name\_\_ == "\_\_main\_\_":

app = CarWashApp()

app.preprocess\_data()

app.show\_menu() Analysis Module (analysis.py)

python

Copy

Edit

import pandas as pd

class CarWashAnalyzer:

def \_init\_(self, df):

self.df = df

def analyze\_churn\_rate(self):

latest = self.df["Transaction Date & Time"].max()

last\_visits = self.df.groupby("Customer ID")["Transaction Date & Time"].max()

churned = (latest - last\_visits).dt.days > 60

churn\_rate = churned.mean()

print(f"\nCustomer Churn Rate: {churn\_rate:.2%}")

churned.name = 'Churned'

churned.to\_csv("churn\_data.csv")

def recommend\_service\_packages(self):

top\_services = self.df["Service Type"].value\_counts().head(3)

print("\nRecommended Service Combos:")

print(top\_services)

top\_services.to\_csv("recommended\_packages.csv", header=["Count"])

def analyze\_seasonality(self):

seasonality = self.df.groupby("Month")["Amount Paid"].sum()

print("\nSeasonality Trends (Revenue by Month):")

print(seasonality)

seasonality.to\_csv("seasonality.csv", header=["Revenue"])

def track\_carbon\_footprint(self):

emission\_factor = 0.05 # kg CO2 per transaction

self.df["CO2\_Emitted"] = emission\_factor

footprint = self.df.groupby("Month")["CO2\_Emitted"].sum()

print("\nMonthly CO2 Emissions (kg):")

print(footprint)

footprint.to\_csv("carbon\_footprint.csv", header=["CO2\_Emitted"])

def identify\_low\_sales\_month(self):

monthly\_sales = self.df.groupby("Month")["Amount Paid"].sum()

low\_month = monthly\_sales.idxmin()

print(f"\nLowest Sales Month: {low\_month} – ₹{monthly\_sales[low\_month]:,.2f}")

monthly\_sales.to\_csv("monthly\_sales.csv", header=["Revenue"])

def rank\_services(self):

ranking = self.df.groupby("Service Type").agg(

Usage\_Count=("Transaction ID", "count"),

Total\_Revenue=("Amount Paid", "sum")

).sort\_values("Total\_Revenue", ascending=False)

print("\nService Ranking by Revenue and Usage:")

print(ranking)

ranking.to\_csv("service\_ranking.csv")

📈 Visualization Module (visualization.py)

python

Copy

Edit

import pandas as pd

import matplotlib.pyplot as plt

import os

class CarWashVisualizer:

def plot\_churn\_rate(self):

data = pd.read\_csv("churn\_data.csv", index\_col=0).squeeze()

counts = data.value\_counts()

counts.index = ["Active", "Churned"]

counts.plot(kind="pie", autopct='%1.1f%%', title="Customer Churn Rate")

plt.ylabel('')

plt.tight\_layout()

plt.show()

def plot\_service\_recommendations(self):

data = pd.read\_csv("recommended\_packages.csv", index\_col=0).squeeze()

data.plot(kind="bar", color="skyblue", title="Top Recommended Services")

plt.xlabel("Service Type")

plt.ylabel("Count")

plt.tight\_layout()

plt.show()

def plot\_seasonality(self):

data = pd.read\_csv("seasonality.csv", index\_col=0).squeeze()

data.plot(kind="line", marker='o', title="Seasonality Trends")

plt.xlabel("Month")

plt.ylabel("Revenue")

plt.grid(True)

plt.tight\_layout()

plt.show()

def plot\_carbon\_footprint(self):

data = pd.read\_csv("carbon\_footprint.csv", index\_col=0).squeeze()

data.plot(kind="bar", color="olive", title="Monthly CO2 Emissions")

plt.xlabel("Month")

plt.ylabel("CO2 (kg)")

plt.tight\_layout()

plt.show()

def plot\_low\_sales\_month(self):

data = pd.read\_csv("monthly\_sales.csv", index\_col=0).squeeze()

low = data.idxmin()

colors = ['red' if m == low else 'lightgrey' for m in data.index]

data.plot(kind="bar", color=colors, title="Monthly Sales (Lowest Highlighted)")

plt.xlabel("Month")

plt.ylabel("Revenue")

plt.tight\_layout()

plt.show()

def plot\_service\_ranking(self):

df = pd.read\_csv("service\_ranking.csv", index\_col=0)

df["Total\_Revenue"].plot(kind="barh", color="seagreen", title="Top Services by Revenue")

plt.xlabel("Revenue")

plt.tight\_layout()

plt.show()

df["Usage\_Count"].plot(kind="barh", color="dodgerblue", title="Top Services by Usage")

plt.xlabel("Usage Count")

plt.tight\_layout()

plt.show()

def show\_all\_plots(self):

print("\nGenerating all available plots...")

if os.path.exists("churn\_data.csv"):

self.plot\_churn\_rate()

if os.path.exists("recommended\_packages.csv"):

self.plot\_service\_recommendations()

if os.path.exists("seasonality.csv"):

self.plot\_seasonality()

if os.path.exists("carbon\_footprint.csv"):

self.plot\_carbon\_footprint()

if os.path.exists("monthly\_sales.csv"):

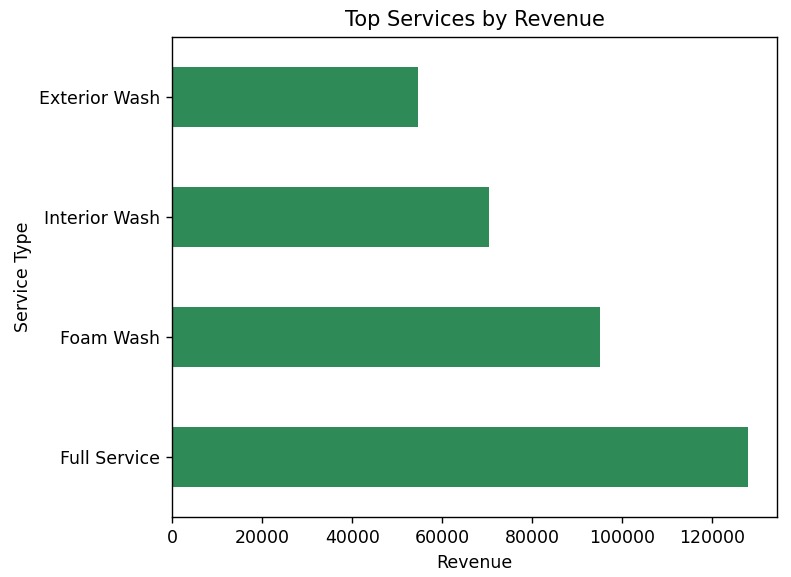
self.plot\_low\_sales\_month()

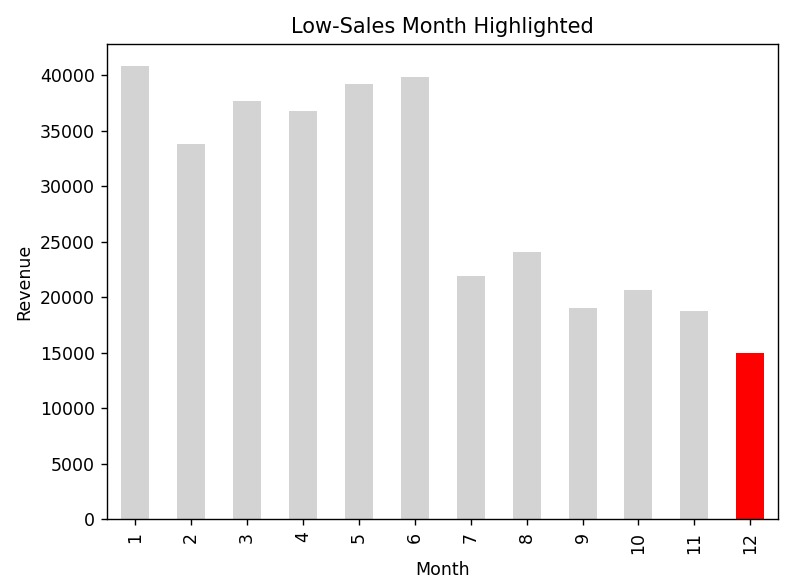
if os.path.exists("service\_ranking.csv"):

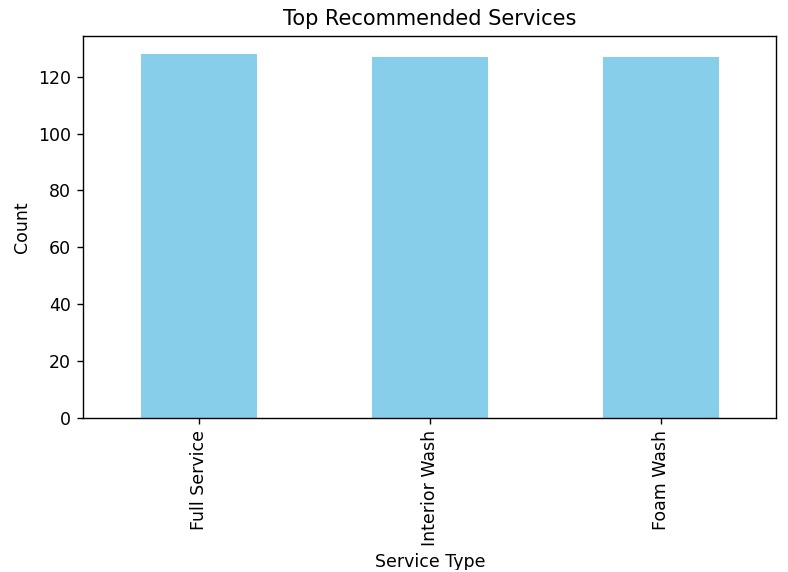
self.plot\_service\_ranking()

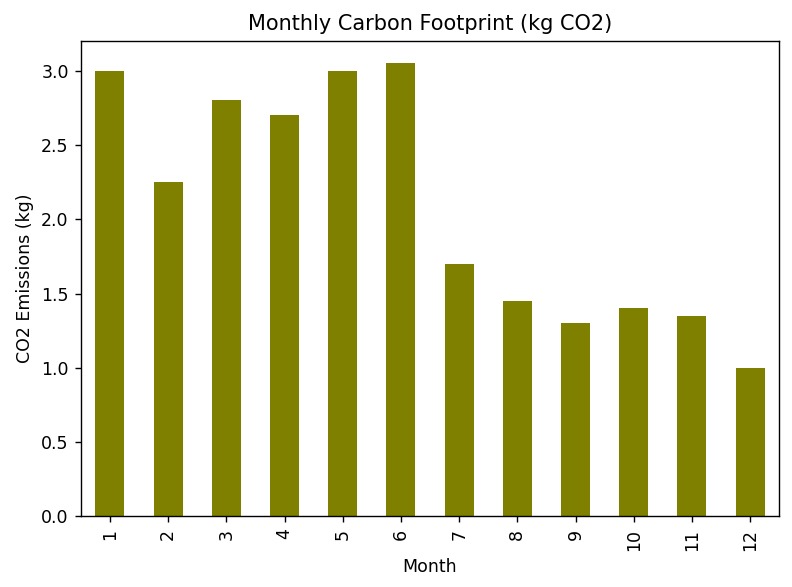
7. Output Screenshots

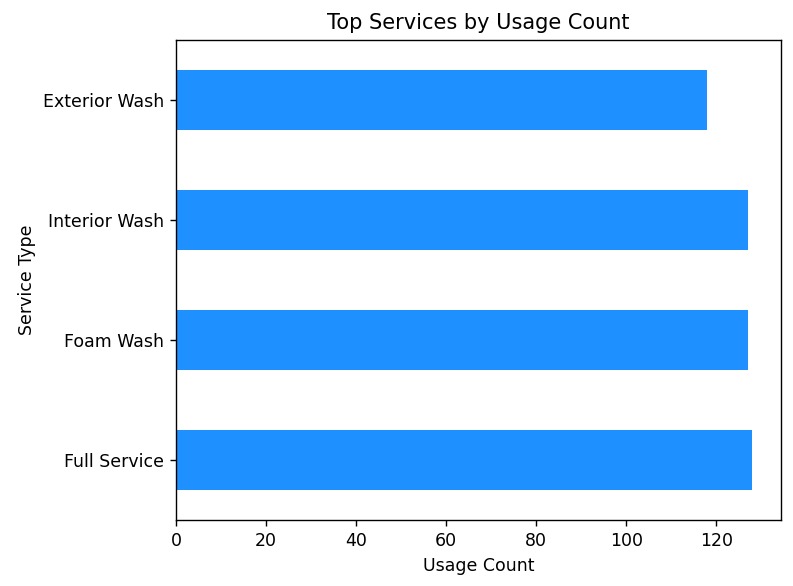
(You can generate these using your app and embed them here.)

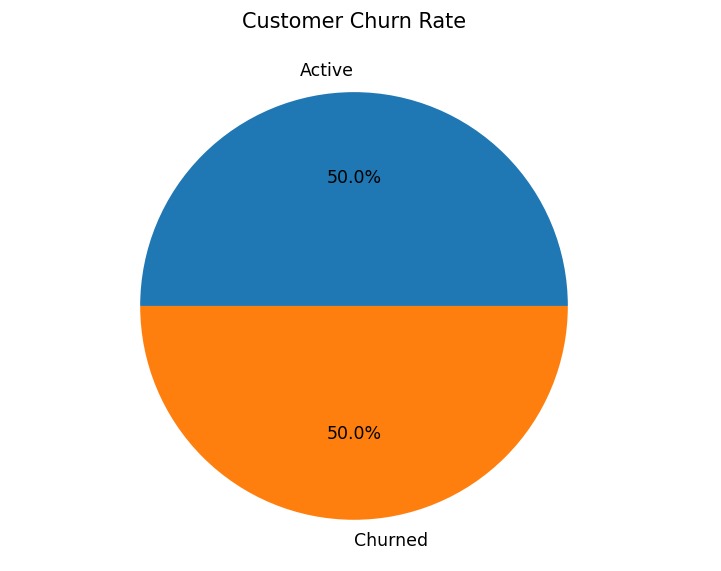


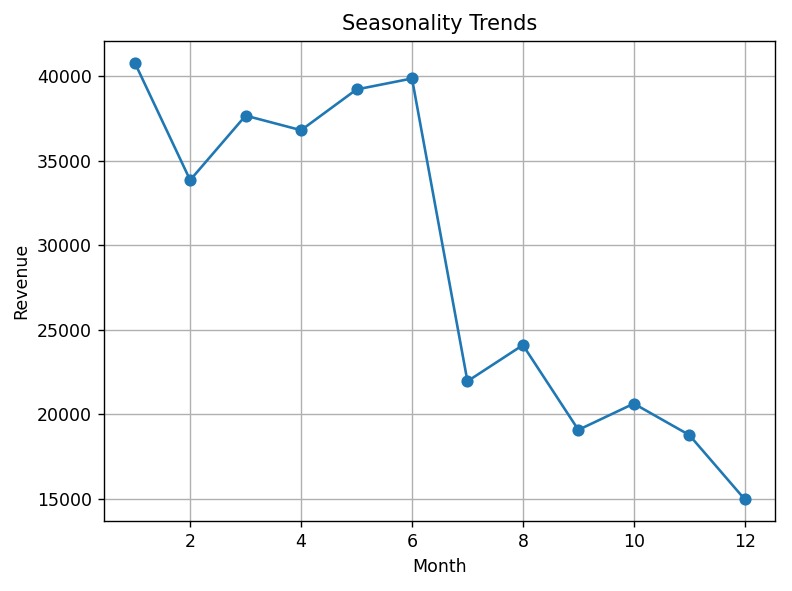












8. Closure

The Car Wash Analytics App helps understand customer behavior, plan promotions, and analyze sales trends. By integrating churn analysis, seasonal trends, and service ranking, it delivers actionable insights with interactive visuals. Future updates could add live customer notification, email-based re-engagement, and real-time dashboards.