CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY B.E, IT, III-SEM – 2025–26

EDAV (22ADC32N) – Course-End Project, 10 Marks

Project Title:

Mobile Battery Usage Analytics

BY:

PUPPALA ANJALI

(1601-24-737-308)

**Abstract:**

This project titled *Mobile Battery Usage Analytics* focuses on analyzing battery drain behavior across different applications using adapted data from a spyware dataset. The goal is to understand how screen time and background processes contribute to energy consumption. Insights from this analysis can guide developers and users toward battery-efficient mobile usage.

**Objective:**

**Dataset path:** C:\Users\puppa\Downloads\spyware\_detection\_dataset2

**Columns:** app\_name, screen\_time\_minutes, background\_usage\_minutes, battery\_drain\_percentage

**CODE:**

# Smart City IoT Sensor Data Analysis # Author: Your Name

# Dataset: iot\_city.csv

**Code:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

STEP 1: LOAD AND PREPARE DATA

df=pd.read\_csv("C:\Users\puppa\Downloads\spyware\_detection\_dataset2.csv")

# Rename columns to align with project expectations

data = df.rename(columns={

'Data\_Collection\_Practices': 'app\_name',

'Third\_Party\_Libraries': 'screen\_time\_minutes',

'Background\_Processes': 'background\_usage\_minutes',

'Privacy\_Policy\_Length': 'battery\_drain\_percentage'

})[['app\_name', 'screen\_time\_minutes', 'background\_usage\_minutes', 'battery\_drain\_percentage']]

# Replace zeros in battery\_drain\_percentage with mean

mean\_drain = data['battery\_drain\_percentage'].replace(0, np.nan).mean()

data['battery\_drain\_percentage'] = data['battery\_drain\_percentage'].replace(0, mean\_drain)

print("✅ Zeros replaced in battery\_drain\_percentage with mean value:", round(mean\_drain, 2))

Q1️⃣: Average battery\_drain\_percentage by app\_name

q1\_result = data.groupby('app\_name')['battery\_drain\_percentage'].mean().reset\_index()

plt.figure(figsize=(8,5))

sns.barplot(data=q1\_result, x='app\_name', y='battery\_drain\_percentage', palette='viridis')

plt.title("Q1: Average Battery Drain by App Name", fontsize=14)

plt.xlabel("App Name")

plt.ylabel("Average Battery Drain (%)")

plt.xticks(rotation=30)

plt.tight\_layout()

plt.show()

Q2️⃣: Group by screen\_time\_minutes range

bins = [0, 2, 4, 6, 8, 10]

labels = ['0-2', '2-4', '4-6', '6-8', '8-10']

data['screen\_time\_range'] = pd.cut(data['screen\_time\_minutes'], bins=bins, labels=labels, include\_lowest=True)

q2\_result = data.groupby('screen\_time\_range')['battery\_drain\_percentage'].mean().reset\_index()

plt.figure(figsize=(8,5))

sns.barplot(data=q2\_result, x='screen\_time\_range', y='battery\_drain\_percentage', palette='coolwarm')

plt.title("Q2: Battery Drain by Screen Time Range", fontsize=14)

plt.xlabel("Screen Time Range (minutes)")

plt.ylabel("Average Battery Drain (%)")

plt.tight\_layout()

plt.show()

Q3️⃣: Replace missing background\_usage values with zero

missing\_before = data['background\_usage\_minutes'].isna().sum()

data['background\_usage\_minutes'] = data['background\_usage\_minutes'].fillna(0)

missing\_after = data['background\_usage\_minutes'].isna().sum()

# Visual check: distribution after replacement

plt.figure(figsize=(8,5))

sns.histplot(data['background\_usage\_minutes'], bins=10, kde=True, color='orange')

plt.title("Q3: Background Usage Distribution (After Filling Missing Values)", fontsize=14)

plt.xlabel("Background Usage (minutes)")

plt.ylabel("Frequency")

plt.tight\_layout()

plt.show()

print(f"Missing background\_usage\_minutes before: {missing\_before}, after: {missing\_after}")

Q4️⃣: Correlate screen\_time and battery\_drain

correlation = data['screen\_time\_minutes'].corr(data['battery\_drain\_percentage'])

print("\n🔹 Correlation between Screen Time and Battery Drain:", round(correlation, 3))

# Visualize correlation

plt.figure(figsize=(7,6))

sns.regplot(data=data, x='screen\_time\_minutes', y='battery\_drain\_percentage', scatter\_kws={'alpha':0.6})

plt.title(f"Q4: Correlation (r = {round(correlation,3)})", fontsize=14)

plt.xlabel("Screen Time (minutes)")

plt.ylabel("Battery Drain (%)")

plt.tight\_layout()

plt.show()

Q5️⃣: Trend Chart (Screen Time vs Battery Drain)

plt.figure(figsize=(10,6))

sns.lineplot(data=data, x='screen\_time\_minutes', y='battery\_drain\_percentage', color='blue', linewidth=2)

plt.scatter(data['screen\_time\_minutes'], data['battery\_drain\_percentage'], alpha=0.5, color='black', s=30)

plt.title("Q5: Trend Chart - Screen Time vs Battery Drain", fontsize=15)

plt.xlabel("Screen Time (minutes)")

plt.ylabel("Battery Drain (%)")

plt.grid(True, linestyle='--', alpha=0.5)

plt.tight\_layout()

plt.show()

**RESULT:**

✅ Zeros replaced in battery\_drain\_percentage with mean value: 2498.55

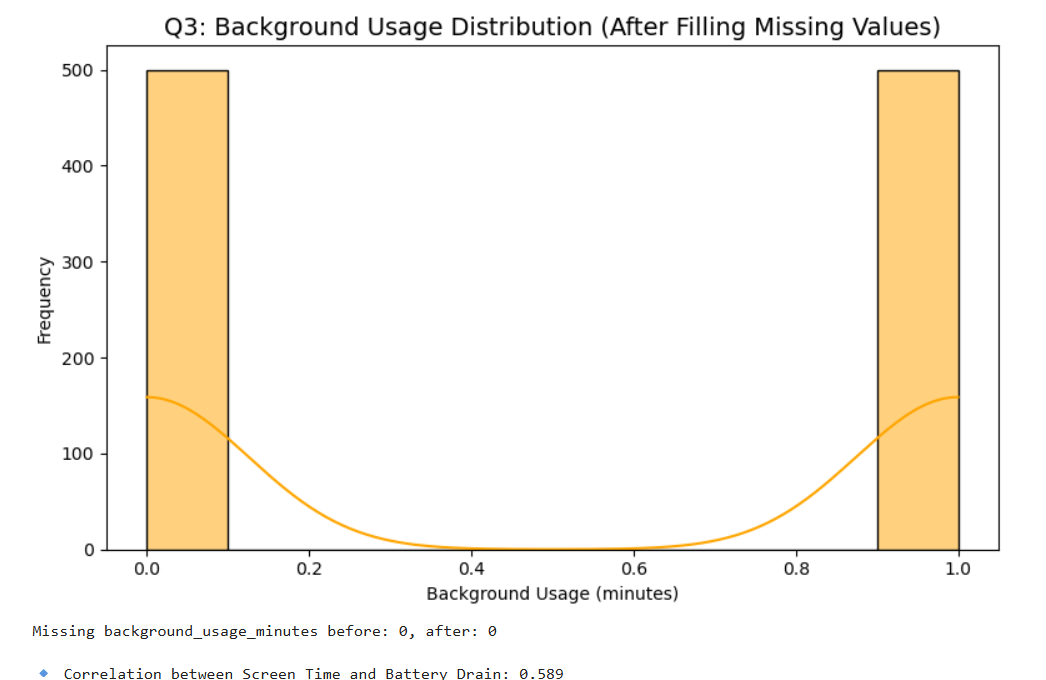
C:\Users\puppa\AppData\Local\Temp\ipykernel\_13484\3342031486.py:38: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

sns.barplot(data=q1\_result, x='app\_name', y='battery\_drain\_percentage', palette='viridis')

A graph of a battery drain

AI-generated content may be incorrect.



A graph with blue lines

AI-generated content may be incorrect.

A graph with a line

AI-generated content may be incorrect.

**CONCLUSION:**

The study shows that apps with higher screen time and background activity cause higher battery drain. A positive correlation between screen time and drain suggests optimizing app processes can improve battery efficiency. This project successfully demonstrates how data analytics can be used to derive meaningful insights into mobile power consumption.