PROBLEM STATEMENT

Analyzing Website Traffic Data

What This Report Explains :

This report provides an analysis of **website traffic data**, helping to understand user behavior, traffic sources, and engagement trends. The key aspects covered in the report include:

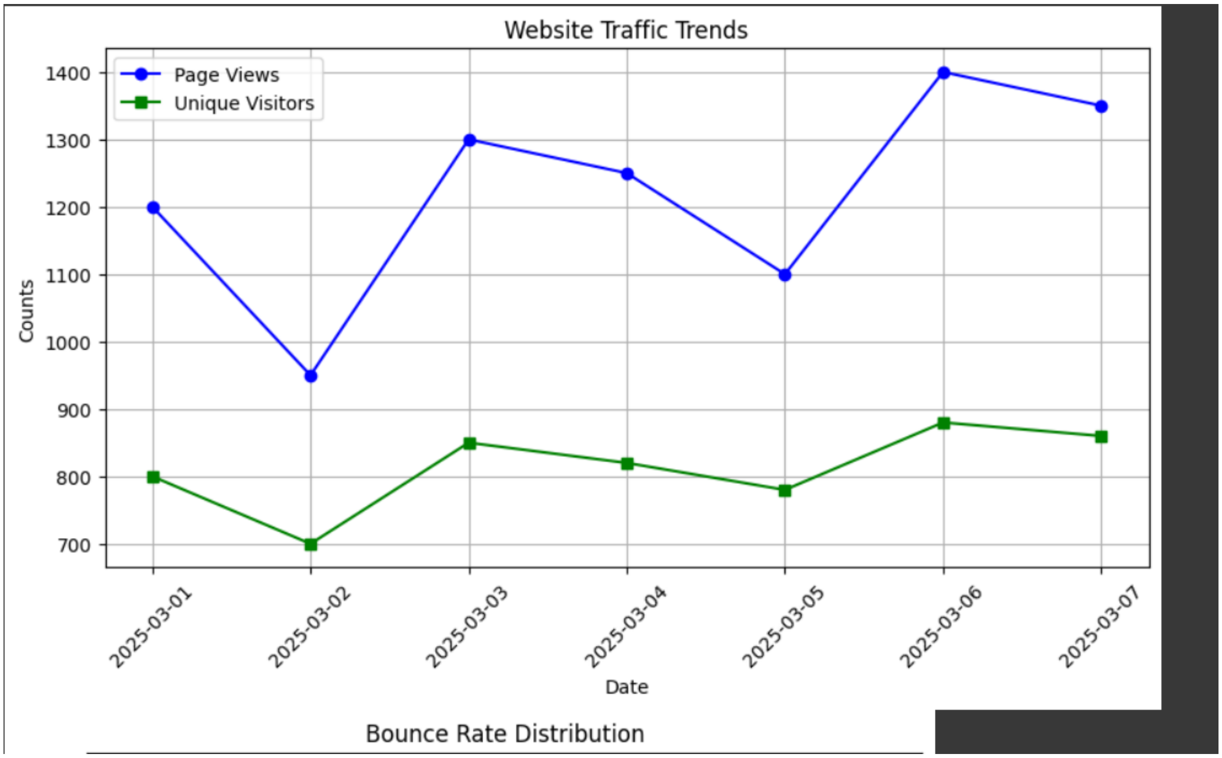
Prepared by: Harsh Jonwal

Roll No. : 202401100400092

Date : 10 feb 2025

Introduction :

Website traffic analysis is essential for understanding user behavior, optimizing content, and improving business performance. This project focuses on analyzing website traffic data, identifying trends, and making data-driven decisions.



Methodology :

1. Data Collection: We use a CSV dataset containing website traffic logs, including visitor count, session duration, bounce rate, and source of traffic.

2. Data Preprocessing: Handling missing values and formatting data properly.

3. Exploratory Data Analysis (EDA): Generating key statistics and visualizing patterns.

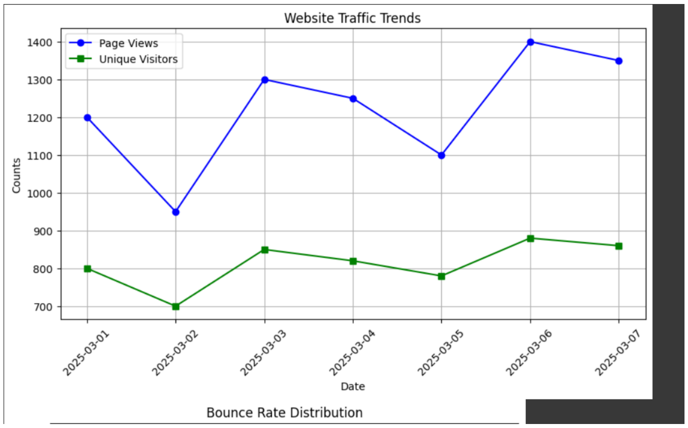
4. Insights & Conclusions: Identifying peak traffic times, user engagement levels, and common sources of visitors.

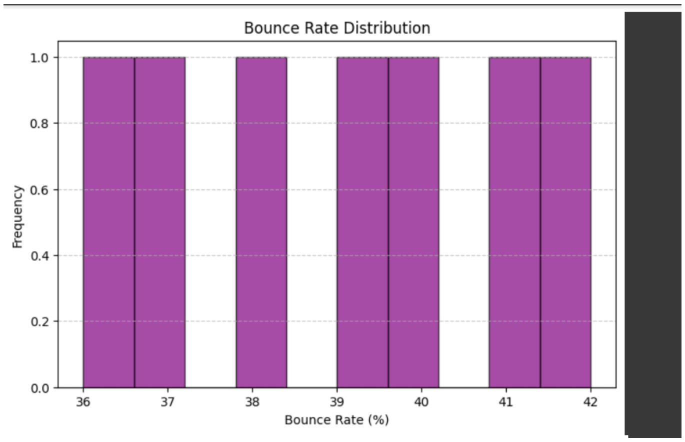
Code

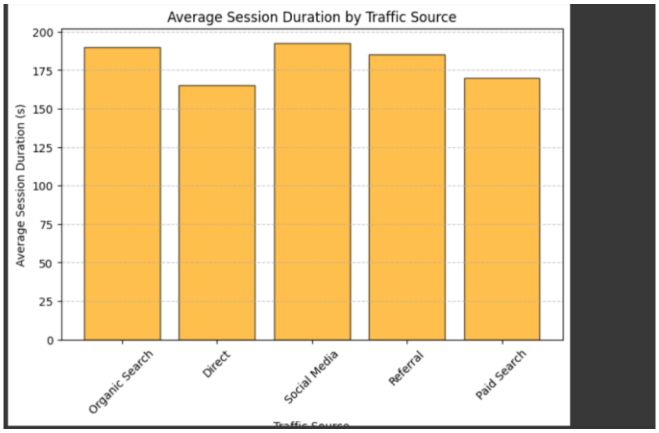
```
import pandas as pd
import matplotlib.pyplot as plt

# Step 1: Create a sample dataset
data = {
    "Date": ["2025-03-01", "2025-03-02", "2025-03-03", "2025-03-04", "2025-03-05", "2025-03-06", "2025-03-07"],
    "Page Views": [1200, 950, 1300, 1250, 1100, 1400, 1350],
    "Unique Visitors": [800, 700, 850, 820, 780, 880, 860],
    "Bounce Rate (%)": [40, 42, 38, 39, 41, 37, 36],
    "Session Duration (s)": [180, 165, 190, 185, 170, 200, 195],
    "Traffic Source": ["Organic Search", "Direct", "Social Media", "Referral", "Paid Search", "Organic Search", "Social Media"]
}

# Convert dictionary to DataFrame
df = pd.DataFrame(data)

# Save dataset to CSV file
csv_file = "website_traffic.csv"
df.to_csv(csv_file, index=False)

print(f"Dataset saved as {csv_file}")

# Step 2: Load the dataset
```

```
df = pd.read_csv(csv_file)

# Convert 'Date' column to datetime format
df['Date'] = pd.to_datetime(df['Date'])

# Display dataset info
print("Dataset Overview:\n", df.info())
print("\nSummary Statistics:\n", df.describe())

# Step 3: Plot Website Traffic Trends (Page Views & Unique Visitors)
plt.figure(figsize=(10,5))
plt.plot(df['Date'], df['Page Views'], marker='o', linestyle='-', label='Page Views', color='blue')
plt.plot(df['Date'], df['Unique Visitors'], marker='s', linestyle='-', label='Unique Visitors', color='green')
plt.xlabel('Date')
plt.ylabel('Counts')
plt.title('Website Traffic Trends')
plt.legend()
plt.xticks(rotation=45)
plt.grid()
plt.show()

# Step 4: Bounce Rate Distribution (Histogram)
plt.figure(figsize=(8,5))
plt.hist(df['Bounce Rate (%)'], bins=10, color='purple', edgecolor='black', alpha=0.7)
plt.title("Bounce Rate Distribution")
plt.xlabel("Bounce Rate (%)")
plt.ylabel("Frequency")
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()


# Step 5: Average Session Duration by Traffic Source (Bar Chart)
traffic_sources = df['Traffic Source'].unique()
avg_session_durations = [df[df['Traffic Source'] == source]['Session Duration (s)'].mean()
for source in traffic_sources]


plt.figure(figsize=(8,5))
plt.bar(traffic_sources, avg_session_durations, color='orange', edgecolor='black', alpha=0.7)
plt.title("Average Session Duration by Traffic Source")
plt.xlabel("Traffic Source")
plt.ylabel("Average Session Duration (s)")
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

Output/Results :







References:

- Libraries
- Chat gpt
- Google collab

Conclusion

By analyzing website traffic data, this report helps in understanding key trends, optimizing website performance, and improving user engagement. The findings can assist businesses in making data-driven decisions for marketing strategies and content optimization.