INTRODUCTION TO AI

WEBSITE TRAFFIC ANALYSIS REPORT

Report File

BACHELOR OF TECHNOLOGY

CSE(AIML)

- NAME: **NITIKA MAAN**
- BRANCH: **CSEAIML-B**
- ROLL NO: **202401100400132**

WEBSITE TRAFFIC ANALYSIS REPORT

Introduction

This report provides an analysis of website traffic data to identify trends and patterns. The dataset includes information on page views, session duration, bounce rate, and traffic sources. Understanding website traffic helps in optimizing user experience and improving site performance.

Methodology

1. Data was collected from the website traffic dataset (`website_traffic.csv`).

2. The dataset was cleaned and processed using Python (Pandas, Matplotlib, Seaborn).

3. Data was visualized using graphs to analyze trends.

4. Insights were derived based on traffic sources, bounce rates, and session duration.

Code

The following Python code was used for the analysis:

```
# Importing necessary libraries

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns


# Creating a sample website_traffic.csv file if it doesn't exist

import os

import random

from datetime import datetime, timedelta
```

```
# Check if the CSV file exists; if not, create it
csv_filename = "website_traffic.csv"
if not os.path.exists(csv_filename):
    # Generate a date range for the last 30 days
    num_days = 30
    start_date = datetime.now() - timedelta(days=num_days)
    dates = [start_date + timedelta(days=i) for i in range(num_days)]

    # Generate sample website traffic data
    data = {
        "timestamp": [date.strftime("%Y-%m-%d %H:%M:%S") for date in dates],
        "page_views": [random.randint(500, 5000) for _ in range(num_days)],
        "session_duration": [random.randint(30, 300) for _ in range(num_days)],  # in seconds
        "bounce_rate": [round(random.uniform(20, 80), 2) for _ in range(num_days)],  # in %
        "traffic_source": random.choices(["Organic", "Direct", "Referral", "Social", "Paid"],
k=num_days)
    }

    # Create and save the dataset
    df = pd.DataFrame(data)
    df.to_csv(csv_filename, index=False)
    print("Sample dataset 'website_traffic.csv' created.")

# Load the dataset
df = pd.read_csv(csv_filename)
```

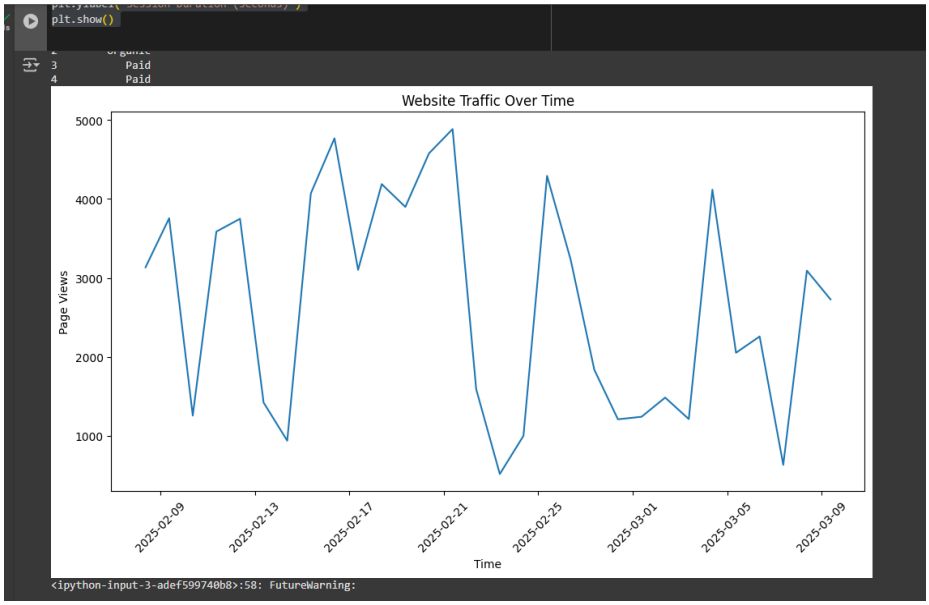
```
# Display basic dataset info
print(df.info())
print(df.head())

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

# Extract hour and day for analysis
df['hour'] = df['timestamp'].dt.hour
df['day'] = df['timestamp'].dt.day_name()

# Plot traffic over time
plt.figure(figsize=(12, 6))
sns.lineplot(x=df['timestamp'], y=df['page_views'])
plt.title("Website Traffic Over Time")
plt.xlabel("Time")
plt.ylabel("Page Views")
plt.xticks(rotation=45)
plt.show()

# Analyze traffic sources
plt.figure(figsize=(8, 4))
sns.countplot(x=df['traffic_source'], data=df, palette="coolwarm")
plt.title("Traffic Source Distribution")
plt.xlabel("Source")
```

```
plt.ylabel("Count")
plt.show()

# Bounce rate analysis
plt.figure(figsize=(8, 4))
sns.histplot(df['bounce_rate'], bins=20, kde=True)
plt.title("Bounce Rate Distribution")
plt.xlabel("Bounce Rate (%)")
plt.ylabel("Frequency")
plt.show()

# Average session duration by day
avg_session_duration = df.groupby('day')['session_duration'].mean()
plt.figure(figsize=(10, 5))
sns.barplot(x=avg_session_duration.index, y=avg_session_duration.values, palette="viridis")
plt.title("Average Session Duration by Day")
plt.xlabel("Day")
plt.ylabel("Session Duration (seconds)")
plt.show()
```

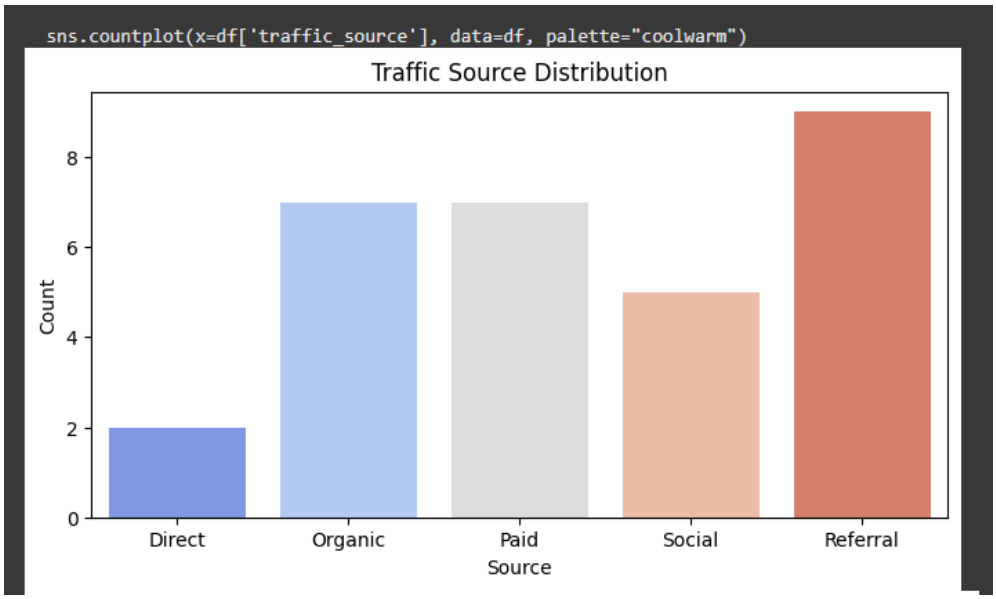
• Output/Results

The following visualizations were generated from the data analysis:

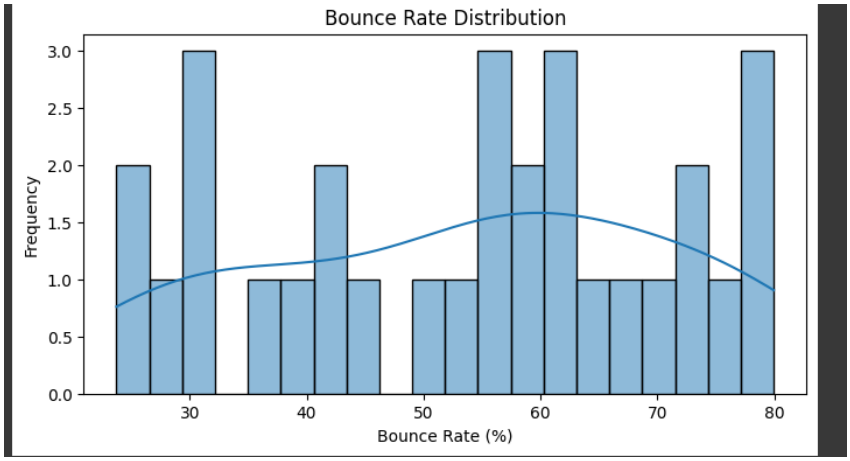
1. Line plot showing website traffic over time.



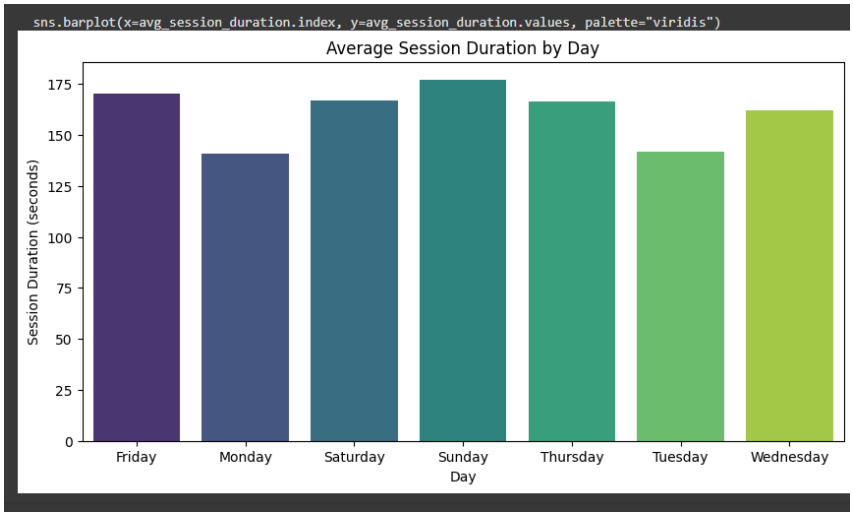
2. Bar chart showing traffic sources.



3. Histogram of bounce rates.



4. Bar chart for average session duration by day.



• References/Credits

Data Source: Simulated website traffic dataset.

Tools Used: Python, Pandas, Matplotlib, Seaborn.

Author:

NITIKA MAAN