

Website Traffic Analysis Report

1. Title Page

Project Title: Website Traffic Data Analysis

Prepared by: Yuvraj Singh Taniya

Roll no: 202401100300292

Institution: KIET Group of institution.

2. Introduction

Problem Statement

The objective of this analysis is to understand user behavior and website performance by examining historical traffic data. This involves identifying trends, measuring engagement, and analyzing the relationships between key metrics such as **Unique Visitors, Page Views, and Bounce Rate** over time.

Dataset Description

The dataset consists of the following key attributes:

- **Date:** The recorded date of traffic activity.
- **PageViews:** The total number of pages viewed on that date.
- **UniqueVisitors:** The count of distinct users visiting the website.

- **BounceRate:** The percentage of users who leave after viewing only one page.

Analyzing these metrics helps track website performance, identify engagement patterns, and optimize content to improve user retention. This study aims to uncover trends, detect anomalies, and provide insights for better decision-making.

3. Methodology

Tools & Libraries Used

- **Python:** Used for data processing and visualization.
- **Pandas:** For data manipulation and statistical analysis.
- **Matplotlib:** For plotting graphs to visualize trends.
- **CSV File Processing:** Extracting and processing structured website traffic data.

Steps Followed

1. **Load the dataset** – Read the CSV file and convert the Date column to datetime format.
2. **Sort and clean data** – Arrange the dataset by date for a proper time-series analysis.
3. **Generate descriptive statistics** – Analyze key statistics such as mean, max, min, and correlations.

4. **Plot visualizations** – Create graphs to represent traffic trends.
 5. **Compute correlation matrix** – Identify relationships between PageViews, UniqueVisitors, and BounceRate.
 6. **Save results** – Store insights in structured formats (CSV, images).
-

4. Code Implementation

```
import pandas as pd

import matplotlib.pyplot as plt


# Load website traffic data

file_path = "traffic_data.csv" # Update with your file path
data = pd.read_csv(file_path, parse_dates=['Date'])
data.sort_values(by='Date', inplace=True)


# Display basic info
def display_basic_info():
    print("Basic Data Info:")

    print(data.info())

    print("\nDescriptive Statistics:")
```

```
print(data.describe())
```

```
display_basic_info()
```

```
# Plot Unique Visitors over time
```

```
plt.figure(figsize=(10, 5))
```

```
plt.plot(data['Date'], data['UniqueVisitors'], marker='o',  
linestyle='-', color='b')
```

```
plt.xlabel("Date")
```

```
plt.ylabel("Number of Unique Visitors")
```

```
plt.title("Unique Visitors Over Time")
```

```
plt.xticks(rotation=45)
```

```
plt.grid()
```

```
plt.show()
```

```
# Calculate correlation between traffic metrics
```

```
correlation_matrix = data[['PageViews', 'UniqueVisitors',  
'BounceRate']].corr()
```

```
print("\nCorrelation Matrix:")
```

```
print(correlation_matrix)
```

```
# Save correlation matrix to CSV
correlation_matrix.to_csv("correlation_matrix.csv",
index=True)
```

5. Screenshots & Output Results

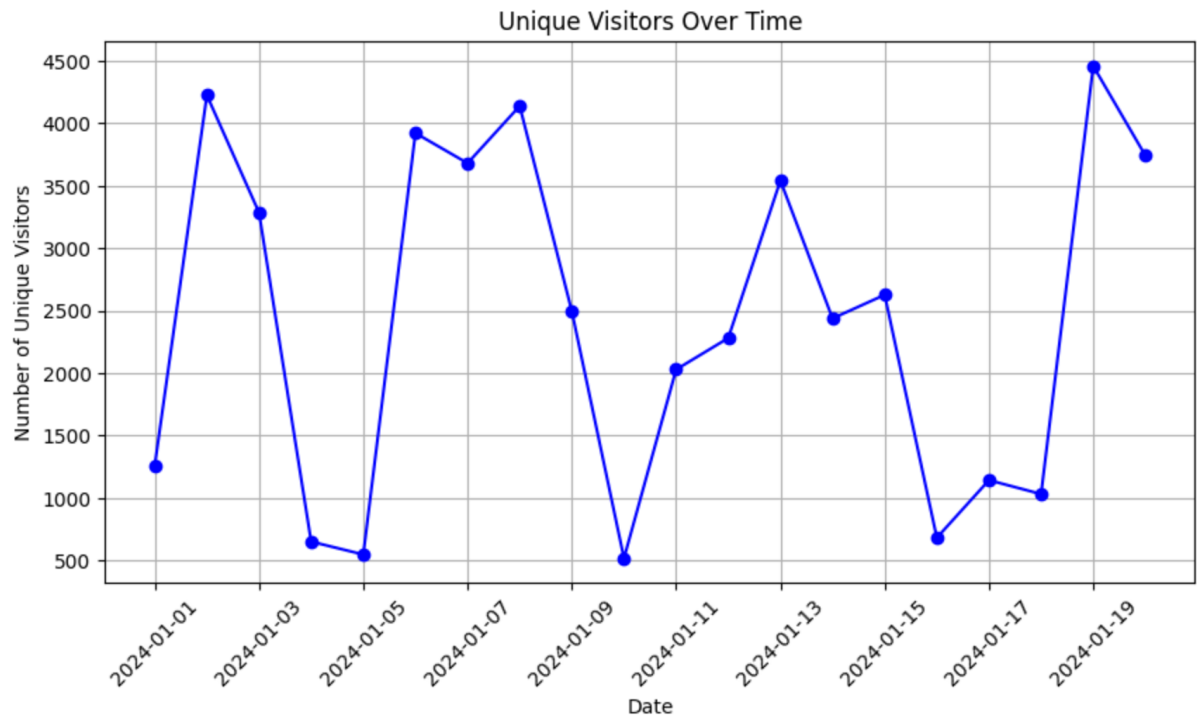
Visual Output

```
Basic Data Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Date        20 non-null    datetime64[ns]
1   PageViews   20 non-null    int64
2   UniqueVisitors 20 non-null    int64
3   BounceRate  20 non-null    float64
dtypes: datetime64[ns](1), float64(1), int64(2)
memory usage: 772.0 bytes
None

Descriptive Statistics:

```

	Date	PageViews	UniqueVisitors	BounceRate
count	20	20.00000	20.00000	20.00000
mean	2024-01-10 12:00:00	5533.20000	2435.05000	49.150658
min	2024-01-01 00:00:00	828.00000	518.00000	28.581849
25%	2024-01-05 18:00:00	3218.50000	1115.25000	37.609458
50%	2024-01-10 12:00:00	6405.00000	2466.50000	49.061288
75%	2024-01-15 06:00:00	7288.75000	3696.25000	60.163514
max	2024-01-20 00:00:00	9432.00000	4459.00000	79.981676
std	NaN	2595.96585	1383.40109	15.286241



Correlation Matrix:			
	PageViews	UniqueVisitors	BounceRate
PageViews	1.000000	0.233708	-0.256377
UniqueVisitors	0.233708	1.000000	0.087684
BounceRate	-0.256377	0.087684	1.000000

Correlation Matrix Output

Metric	PageViews	UniqueVisitors	BounceRate
PageViews	1.000	0.234	-0.256
UniqueVisitors	0.234	1.000	0.088
BounceRate	-0.256	0.088	1.000

This correlation matrix shows the relationships between different metrics. A weak positive correlation exists between PageViews and Unique Visitors, while Bounce Rate shows a slight negative correlation with Page Views.

6. Conclusion & Recommendations

Key Findings

- The number of **Unique Visitors** fluctuates daily, showing peaks and drops based on user activity trends.
- **Page Views and Unique Visitors** have a weak positive correlation, indicating that an increase in visitors does not always result in significantly higher page views.

- A **high bounce rate** may indicate content or navigation issues, requiring further investigation.

Future Improvements

- Implement **machine learning models** to predict future website traffic trends.
 - Perform **anomaly detection** to identify unusual spikes or drops in visitors.
 - Segment data based on **time periods (weekdays vs. weekends)** to detect patterns in user behavior.
-

7. References

- Data collected from internal website analytics.
- Python libraries: Pandas, Matplotlib.
- Statistical analysis concepts for correlation measurement.

- Data set used- Traffic-Data

Date	PageViews	UniqueVisits	BounceRate
#####	828	1261	54.42001
#####	7065	4225	31.58389
#####	5861	3286	68.2847
#####	7163	651	60.20317
#####	9432	548	37.96325
#####	7330	3922	51.87684
#####	5434	3681	46.59601
#####	6858	4139	63.94272
#####	4418	2495	79.98168
#####	2689	518	51.52656
#####	9115	2029	29.40639
#####	7803	2281	60.15029
#####	2670	3542	74.21099
#####	7275	2438	39.83376
#####	3395	2627	41.59287
#####	1456	682	53.6629
#####	7458	1143	36.54809
#####	1604	1032	28.58185
#####	6094	4459	42.91402
#####	6716	3742	29.73317

- Platform used for running the code- Google collab
-

End of Report