Website Traffic Analysis

|  |  |
| --- | --- |
| Team members | Name |
| Team member 1 | Lokesh S |
| Team member 2 | Sanjay K |
| Team member 3 | Tamil selvan J |
| Team member 4 | Jerry utoes X |

|  |  |
| --- | --- |
| Department | Ai and ds |
| College name | Adhi college of engineering and technology |

|  |  |
| --- | --- |
| Domain name | Data analytics with cognos |

Problem Statement:

The project aims to perform an analysis of website traffic data to gain insights into user engagement, such as page loads, unique visits, first-time visits, and returning visits. The goal is to understand how users interact with the website over time and identify patterns and trends.

Understanding:

Website traffic analysis is essential for businesses and website owners to make data-driven decisions, improve user experience, and optimize their online presence. By analyzing traffic data, we can answer questions such as:

1.How many users visit the website daily, and how has this trend changed over time?

2.What is the distribution of page loads per day?

3.How many visitors are unique, and how many are returning?

4.Are there any spikes or dips in traffic on specific days or during certain periods?

About the Dataset:

The dataset provided for this project contains the following columns:

'Row': An index or unique identifier for each record.

'Day': The day of the week.

'Day.Of.Week': The numerical representation of the day of the week.

'Date': The date of the data point.

'Page.Loads': The number of page loads on that date

'Unique.Visits': The count of unique visitors.

'First.Time.Visits': The count of first-time visitors.

'Returning.Visits': The count of returning visitors.

Implementation Technique:

Python is used as the programming language.

Pandas is utilized for data manipulation and analysis.

Matplotlib is employed for data visualization.

The code follows best practices for data analysis, including data cleaning, aggregation, and visualization.

Data Preprocessing:

Data preprocessing is a crucial step in any data analysis project. It involves cleaning and preparing the dataset to make it suitable for analysis. In your project, the following data preprocessing steps were performed:

Loading Data: The first step is to load the dataset from a CSV file into a Pandas DataFrame. This is done using the pd.read\_csv('your\_dataset.csv') function, where 'your\_dataset.csv' should be replaced with the actual file path to your dataset.

Converting Date Column: The 'Date' column, which contains date information, is converted to a datetime object using pd.to\_datetime(data['Date']). This conversion allows for easier date-based operations and sorting.

Sorting Data: Sorting the data by date is an optional but often helpful step. It ensures that the data is arranged chronologically, making it easier to analyze time-based trends. Sorting is performed using data.sort\_values(by='Date', inplace=True).

Data Analysis:

Data analysis is the process of extracting meaningful insights and information from your dataset. In your project, the following data analysis steps were performed:

Grouping and Aggregation: To calculate daily statistics, the data is grouped by the 'Date' column using data.groupby('Date'). The agg function is then used to aggregate the data within each group. In this case, it sums the 'Page.Loads', 'Unique.Visits', 'First.Time.Visits', and 'Returning.Visits' columns for each date.

Storing Results: The aggregated results are stored in the daily\_stats DataFrame, which contains daily totals for page loads, unique visits, first-time visits, and returning visits.

Visualization:

Data visualization is a powerful way to convey insights from your data. In your project, Matplotlib is used for creating visualizations. Here's how the data is visualized:

Creating a Figure: A figure is created using plt.figure(figsize=(12, 6)), specifying the size of the figure for better presentation.

Line Plot: Four line plots are created using plt.plot() for each of the four metrics: page loads, unique visits, first-time visits, and returning visits. The x-axis represents dates, and the y-axis represents the count of each metric.

Labels and Title: Labels for the x-axis and y-axis are added using plt.xlabel() and plt.ylabel(). A title for the plot is set using plt.title().

Legend: A legend is included using plt.legend() to distinguish between the different lines on the plot.

Grid Lines: Grid lines are added for better readability using plt.grid(True).

Displaying the Plot: Finally, the plot is displayed using plt.show().