**WEBSITE TRAFFIC ANALYSIS**

**OBJECTIVE**

The Website Traffic Analysis project aims to analyze website traffic data to understand user behaviour, popular pages, and traffic sources. This analysis will help website owners improve the user experience and make data-driven decisions.

**STEPS FOR IMPLEMENTATION**

**Step 1: Data Collection**

* Identify the data sources you want to collect information from (e.g., website logs, Google Analytics).
* Set up data collection tools or scripts to capture website traffic data.
* Ensure that you comply with data privacy regulations when collecting user data.

**Step 2: Data Preprocessing**

* Extract data from the chosen data sources.
* Handle missing data by imputing, removing, or marking missing values.
* Remove duplicate records from the dataset.
* Ensure data types are consistent and appropriate for analysis.
* Document the data cleaning process for transparency.

**Step 3: ETL (Extract, Transform, Load)**

* Develop ETL pipelines to regularly extract data from the data source.
* Transform the data to fit the database schema, including data type conversions and aggregations.
* Load the transformed data into the database.
* Automate the ETL process to ensure regular data updates.

**Step 4: Data Visualization**

* Select appropriate data visualization tools and libraries (e.g., Matplotlib, Seaborn, Tableau).
* Create interactive dashboards and reports to visualize website traffic metrics, such as page views, unique visits, and traffic sources.

**SOURCE CODE**

import pandas as pd

data = pd.read\_csv('daily-website-visitors.csv')

# Inspect the data

print(data.head())

print(data.dtypes)

# Cleaning Setups Removing the numeric columns

numeric\_cols = ["Page.Loads", "Unique.Visits", "First.Time.Visits", "Returning.Visits"]

data[numeric\_cols] = data[numeric\_cols].replace({',': ''}, regex=True).astype(int)

# Convert the "Date" column to a datetime object

data["Date"] = pd.to\_datetime(data["Date"], format='%m/%d/%Y')

data.to\_csv('cleaned\_data.csv', index=False)

# visualization

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Read the cleaned data

df = pd.read\_csv('cleaned\_data.csv')

# Basic statistics and overview of the data

print(df.describe())

# Plotting Page Loads over Date

plt.figure(figsize=(12, 6))

sns.lineplot(data=df, x='Date', y='Page.Loads')

plt.title('Page Loads Over Time')

plt.xlabel('Date')

plt.ylabel('Page Loads')

plt.xticks(rotation=45)

plt.show()

# Plotting Unique Visits over Date

plt.figure(figsize=(12, 6))

sns.lineplot(data=df, x='Date', y='Unique.Visits')

plt.title('Unique Visits Over Time')

plt.xlabel('Date')

plt.ylabel('Unique Visits')

plt.xticks(rotation=45)

plt.show()

# Pairplot to visualize relationships between numerical columns

sns.pairplot(df, diag\_kind='kde')

plt.show()

**OUTPUT**

Row Day.Of.Week Page.Loads Unique.Visits \

count 2167.000000 2167.000000 2167.000000 2167.000000

mean 1084.000000 3.997231 4116.989386 2943.646516

std 625.703338 2.000229 1350.977843 977.886472

min 1.000000 1.000000 1002.000000 667.000000

25% 542.500000 2.000000 3114.500000 2226.000000

50% 1084.000000 4.000000 4106.000000 2914.000000

75% 1625.500000 6.000000 5020.500000 3667.500000

max 2167.000000 7.000000 7984.000000 5541.000000

First.Time.Visits Returning.Visits

count 2167.000000 2167.000000

mean 2431.824181 511.822335

std 828.704688 168.736370

min 522.000000 133.000000

25% 1830.000000 388.500000

50% 2400.000000 509.000000

75% 3038.000000 626.500000

max 4616.000000 1036.000000





