WEBSITE TRAFFIC ANALYSIS

PHASE-5

PROJECT SUMMARY:

The project is based on the website traffic analysis that uses machine learning models analysis using the dataset and produce result about the website traffic.

AIM:

To create a clear objective, collecting and storing data, visualizing it using tools like IBM Cognos, and enhancing the process with Python for data preprocessing, custom analytics, and automation. This integrated approach allows you to gain valuable insights and make informed decisions about your website's performance and user engagement.

Here's an overview of the objectives, data collection process, data visualization using IBM Cognos, and Python code integration for website traffic analysis:

OBJECTIVES:

1. USER BEHAVIOR ANALYSIS: Understand how users interact with your website, such as page views, clicks, and time spent on each page.

2. TRAFFIC SOURCES: Identify the sources of web traffic, including organic search, direct traffic, referral traffic, and social media.

3. CONVERSION TRACKING: Monitor conversion goals like form submissions, product purchases, or any other desired actions.

4. PERFORMANCE MONITORING: Analyze website performance, including page load times, errors, and user engagement.

5. CONTENT OPTIMIZATION: Identify popular content and areas for improvement to enhance the user experience.

DATA COLLECTION PROCESS:

1. WEB ANALYTICS TOOLS: Implement a web analytics tool like Google Analytics or Matomo (formerly Piwik) to collect data about your website's traffic. These tools provide tracking codes that you embed on your web pages.

2. DATA STORAGE: The collected data is usually stored in a database for further analysis. You can use databases like MySQL, PostgreSQL, or cloud-based solutions like Google BigQuery.

3. DATA PREPARATION: The collected data may need to be cleaned and transformed into a format suitable for analysis. This can be done using tools like Python, pandas, or ETL (Extract, Transform, Load) processes.

DATA VISUALIZATION USING IBM COGNOS:

IBM Cognos is a business intelligence and data visualization tool. You can use it to create interactive and insightful dashboards and reports for website traffic analysis. Here's a high-level process:

1. DATA CONNECTION: Connect IBM Cognos to your data source, which can be a database containing website traffic data.

2. DATA MODELING: Define a data model within IBM Cognos to represent the structure of your data. Create dimensions, measures, and hierarchies as needed.

3. REPORT DESIGN: Design reports and dashboards to visualize website traffic data. You can create bar charts, line graphs, tables, and more to represent key metrics and trends. These visuals can showcase data like page views, user demographics, traffic sources, and conversion rates.

4. INTERACTIVITY: Configure filters, drill-through capabilities, and interactive elements in your reports to allow users to explore the data.

5. SCHEDULE AND SHARE: Set up automated report generation and distribution to key stakeholders.

PYTHON CODE INTEGRATION:

To enhance your website traffic analysis, you can integrate Python into the process:

1. DATA PREPROCESSING: Use Python and libraries like pandas to preprocess and clean the data before it's imported into IBM Cognos. You can also perform advanced data transformations and calculations in Python.

2. MACHINE LEARNING: Python can be used to apply machine learning models to predict future website traffic trends or user behavior.

3. CUSTOM ANALYTICS: For specialized analysis or statistical tests not supported by IBM Cognos, you can write custom Python code and integrate the results into your reports.

4. AUTOMATION: Use Python scripts to automate tasks, such as fetching data from your web analytics tool, updating the database, or scheduling report generation.

Integration between IBM Cognos and Python can be achieved through various means, such as calling Python scripts from within Cognos or using Python libraries for data manipulation and analysis alongside Cognos.

Website traffic analysis involves setting

Website traffic analysis provides valuable data and insights that can help website owners improve user experience in several ways. By understanding how users interact with their website, owners can make data-driven decisions to enhance usability, content, and overall user satisfaction. Here are some keyways in which website traffic analysis can aid in this process:

1. USER BEHAVIOR ANALYSIS:

Website analytics tools can track user behavior, including pages visited, time spent on each page, and click-through rates. By analyzing this data, website owners can identify popular content and navigation paths. This information can guide them in optimizing the user journey, improving the layout of their site, and promoting relevant content more effectively.

2. CONTENT OPTIMIZATION:

Insights from website traffic analysis can help identify which pieces of content resonate with users and which ones may need improvement or removal. Website owners can use this information to refine their content strategy, create more of what users find valuable, and ensure their content aligns with users' interests and needs.

3. CONVERSION RATE OPTIMIZATION (CRO):

Website owners can use traffic data to analyze conversion funnels and identify where users drop off in the conversion process. By pinpointing areas where users are abandoning their journey, website owners can make targeted improvements to reduce friction, simplify forms, and enhance calls to action. This can lead to higher conversion rates, whether it's making a purchase, signing up for a newsletter, or any other desired action.

4. MOBILE-FRIENDLY OPTIMIZATION:

Website traffic analysis often includes device and browser information. Website owners can use this data to determine the prevalence of mobile users and ensure their website is responsive and mobile-friendly. Improving mobile user experience is crucial, as more and more users access websites via smartphones and tablets.

5. PAGE LOAD SPEED:

Slow-loading pages can negatively impact user experience and lead to higher bounce rates. Website traffic analysis tools can provide data on page load times. Owners can use this information to identify slow-loading pages and optimize them for faster performance, resulting in a smoother user experience.

6. A/B TESTING:

Website owners can run A/B tests based on insights from traffic analysis. By testing different variations of pages, elements, or content, they can determine what resonates best with their audience. A/B testing allows for data-driven decisions to refine the website continuously.

7. SEARCH ENGINE OPTIMIZATION (SEO):

Website traffic data can show which keywords and search queries drive the most organic traffic. Website owners can use this information to optimize their content for search engines, helping users find their website more easily. This enhances the overall user experience by ensuring users can access the content they seek.

8. USER DEMOGRAPHICS AND LOCATION:

Analyzing user demographics and location data can help tailor content and user experiences to specific segments of the audience. Website owners can provide more relevant content, language options, and localized information, enhancing user engagement.

9. ERROR TRACKING:

By monitoring error pages and broken links, website owners can improve user experience by ensuring a seamless, frustration-free navigation experience. Fixing broken links and resolving errors promptly is essential for maintaining user trust.

In summary, website traffic analysis is a powerful tool that enables website owners to gain deep insights into user behavior, preferences, and pain points. By acting on these insights, website owners can continuously refine their websites, creating a more user-friendly and engaging experience, resulting in higher user satisfaction and potentially increased conversions.

Replicating website traffic analysis and generating visualizations using IBM Cognos and Python involves several steps. IBM Cognos is a business intelligence and data analytics tool, while Python will be used for data preprocessing and custom analysis. Below are the steps to achieve this, along with example outputs of visualizations and analyses:

STEP 1: DATA COLLECTION

Collect website traffic data from various sources, such as web server logs or Google Analytics. Ensure the data includes relevant information like timestamps, page views, user locations, and any other metrics you want to analyze.

STEP 2: DATA PREPROCESSING WITH PYTHON

Use Python to preprocess the data. You might need to clean, transform, and aggregate the data before feeding it into IBM Cognos for visualization. Here's an example using Python:

PYTHON CODE:

Importing libraries:

```
import pandas as pd
```

Load website traffic data:

```
data = pd.read_csv('website_traffic_data.csv')
```

Perform data cleaning and transformation as needed

Example: Convert timestamps to datetime objects:

```
data['timestamp'] = pd.to_datetime(data['timestamp'])
```

Aggregate data by day:

```
daily_data = data.groupby(data['timestamp'].dt.date).sum()
```

Export preprocessed data:

```
daily_data.to_csv('preprocessed_website_data.csv', index=False)
```

STEP 3: CREATE A DATA SOURCE IN IBM Cognos

1. Open IBM Cognos.
2. Create a new data source connection to the preprocessed data (e.g., CSV file).
3. Define data items and set their data types (e.g., date, integer) in the data source.

STEP 4: BUILD REPORTS AND DASHBOARDS IN IBM Cognos

Create reports and dashboards in IBM Cognos using the data source you defined. Here are some example visualizations you can create:

1. Line Chart for Daily Page Views: Show the trend of daily page views over time.

2. Geo Map for User Locations: Visualize user locations on a map.

3. Bar Chart for Most Visited Pages: Display the most visited pages on your website.

STEP 5: DATA ANALYSIS WITH PYTHON (OPTIONAL)

If you want to perform custom analyses that are not easily achievable in IBM Cognos, you can use Python. For example, you can analyze the correlation between page load times and bounce rates, or calculate average session duration. Here's an example:

PYTHON CODE

Calculate average session duration:

```
daily_data['session_duration'] = daily_data['total_time_on_site'] / daily_data['total_sessions']
```

Analyze the correlation between page load times and bounce rates:

```
correlation = daily_data['page_load_time'].corr(daily_data['bounce_rate'])

print("Correlation between page load time and bounce rate:", correlation)
```

STEP 6: EXPORT AND SHARE RESULTS

Export and share your reports and analyses from IBM Cognos. You can export them as PDFs, Excel files, or other formats for sharing with stakeholders.

STEP 7: SCHEDULE AND AUTOMATE

Consider automating this process by scheduling regular data updates and report generation in IBM Cognos and Python. This ensures that you have up-to-date insights into your website traffic.

Remember that the example outputs and visualizations are based on the data and metrics you collect and the specific analyses you want to perform. Adjust the steps and visualizations as needed to match your website traffic analysis requirements.

EXAMPLE:

SEVERAL HELPFUL PACKAGES TO LOAD:

```
import math
from scipy.stats import norm
```

LINEAR ALGEBRA:

```
import numpy as np
```

DATA PROCESSING:

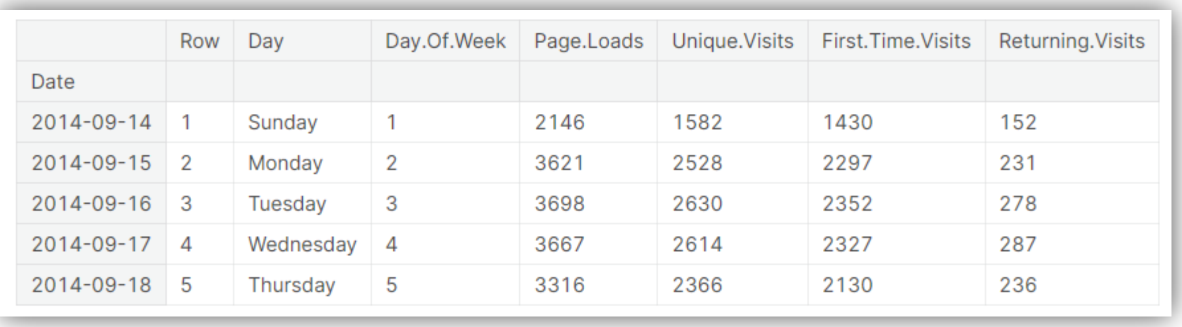
```
import pandas as pd # data processing
import matplotlib.pyplot as plt
from IPython.core.display import HTML
```

INPUT DATA FILES:

```
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

READ CSV:

```
df = pd.read_csv("/kaggle/input/daily-website-visitors/daily-website-visitors.csv", \
            index_col = 'Date', thousands = ',', parse_dates=True)
df.head()
```

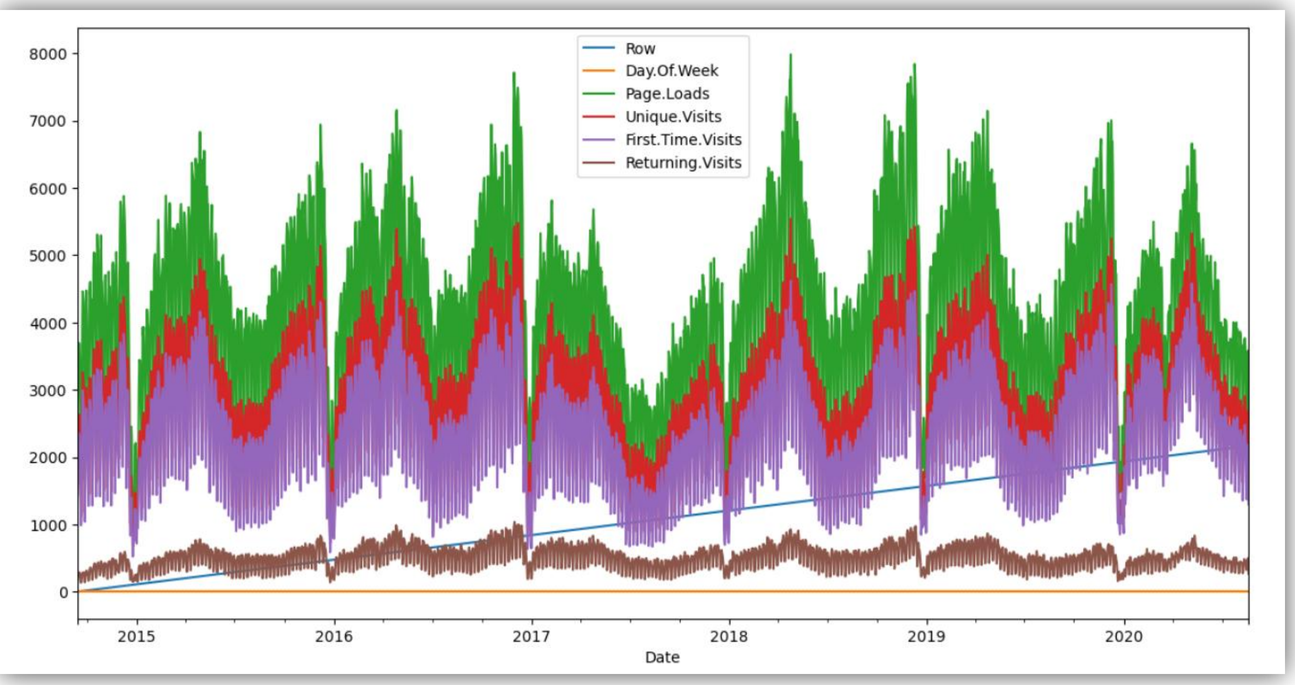


	Row	Day	Day.Of.Week	Page.Loads	Unique.Visits	First.Time.Visits	Returning.Visits
Date							
2014-09-14	1	Sunday	1	2146	1582	1430	152
2014-09-15	2	Monday	2	3621	2528	2297	231
2014-09-16	3	Tuesday	3	3698	2630	2352	278
2014-09-17	4	Wednesday	4	3667	2614	2327	287
2014-09-18	5	Thursday	5	3316	2366	2130	236

PLOT GRAPH:

```
df.plot(figsize=(14,7))
```

```
<AxesSubplot:xlabel='Date'>
```



PRINT(px) AND PRINT(py):

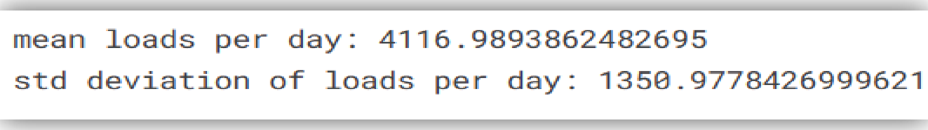
```
def prob(t, n, lmbda):
   return math.pow(lmbda * t, n)/math.factorial(n)*math.exp(-lmbda*t)

mean = df['Page.Loads'].mean()
print( "mean loads per day:", mean)

std = df['Page.Loads'].std()
print( "std deviation of loads per day:", std)

n = 1
px = np.linspace(1, 8000, 50)
py = np.zeros(50)
for i in range(0, 50):
   x = (px[i]-mean)/std
```

```
    p = norm.pdf(x)
    py[i] = 1000*p
```



```
mean loads per day: 4116.9893862482695
std deviation of loads per day: 1350.9778426999621
```

FIGURE:

```
fig, ax1 = plt.subplots()
df['Page.Loads'].plot.hist(ax = ax1, label='Page.Loads')

plt.plot([mean, mean], [0, 480], label='mean')

plt.plot(px, py, label='normal', color='red')
plt.legend()
plt.show()

HTML('<h3>using normal approximation to binomial distribution</h3>')
```

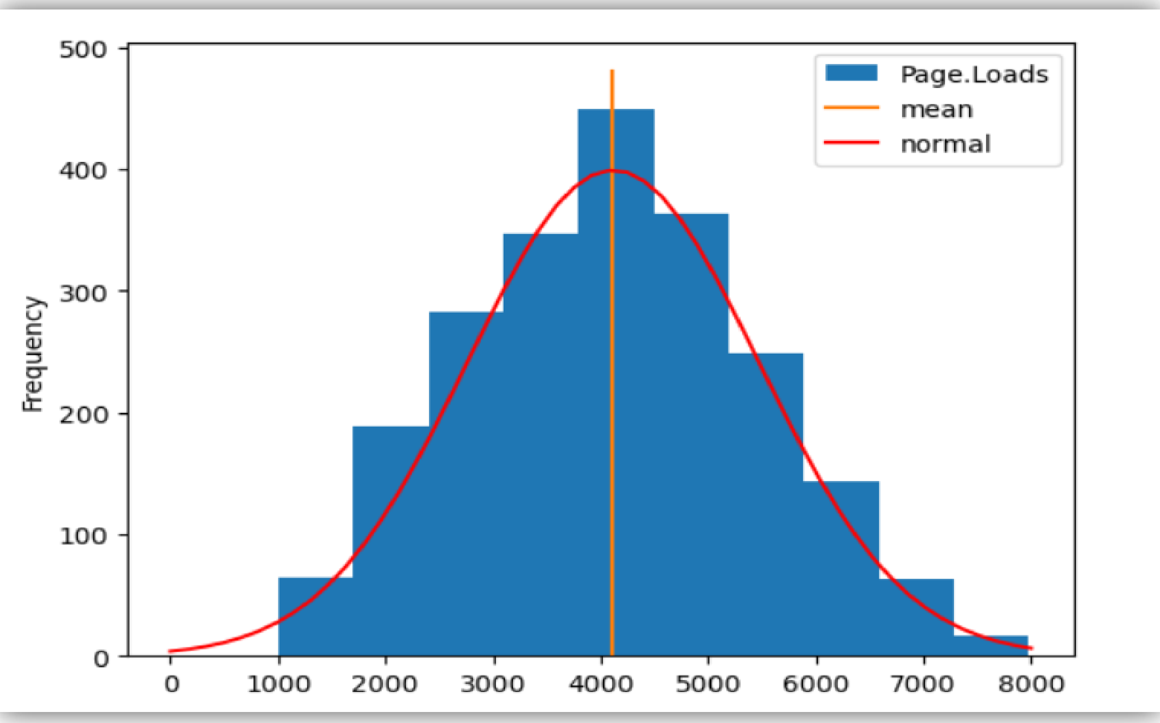
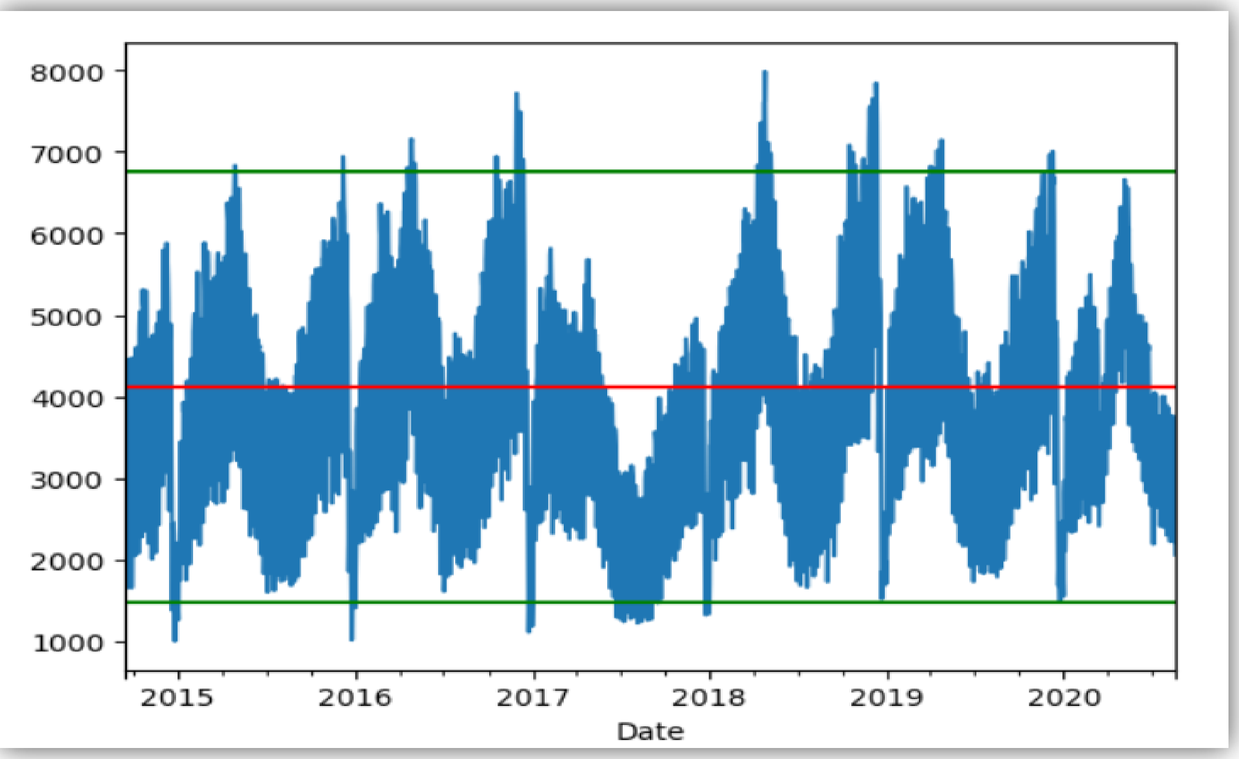


FIGURE:

```
fig, ax1 = plt.subplots()
df['Page.Loads'].plot(ax = ax1, label='Page.Loads')
plt.plot([df.index[0], df.index[-1]], [mean, mean], color='red')
upper = mean + 1.96*std
lower = mean - 1.96*std
plt.plot([df.index[0], df.index[-1]], [upper, upper], color='green')
plt.plot([df.index[0], df.index[-1]], [lower, lower], color='green')
plt.show()
```

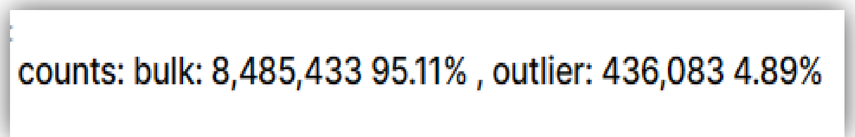


COUNT AND OUTLINE:

```
outlier = 0
bulk = 0
for v in df['Page.Loads']:
    if v > upper or v < lower:
        outlier = outlier + v
    else:
        bulk = bulk + v

tot = bulk + outlier
pctbulk = bulk / tot * 100.0
pctout = outlier / tot * 100.0
```

```
HTML("<h4>counts: bulk: {:,} {:.2f}% , outlier: {:,} {:.2f}%</h4>".format(bulk, pctbulk, outlier,
pctout))
```



```
counts: bulk: 8,485,433 95.11% , outlier: 436,083 4.89%
```

CONCLUSION:

website traffic analysis with IBM Cognos and Python integration provides a comprehensive approach to understanding user behavior, enabling data-driven decisions for improving the overall user experience, driving engagement, and achieving business goals.