Data Analytics with Cognos

WEBSITE TRAFFIC ANALYSIS

GROUP X

1. S. Sasi Priya
2. V. Ramya
3. G. Thirisha
4. R. Megha

Performing more complex analyses on website traffic data using Python libraries like Pandas and Matplotlib typically involves loading and preprocessing the data, conducting time series analysis, user segmentation, and potentially building machine learning models for predictions. Below, I'll provide a step-by-step guide on how to approach these tasks using sample data. Keep in mind that you'll need to adapt this code to your specific dataset.

Step 1: Load and Preprocess Data

```
import pandas as pd

# Load your website traffic data into a Pandas DataFrame

df = pd.read_csv('website_traffic_data.csv')

# Ensure the 'timestamp' column is in DateTime format

df['timestamp'] = pd.to_datetime(df['timestamp'])

# Set 'timestamp' as the index

df.set_index('timestamp', inplace=True)

# Resample data to daily frequency for time series analysis

daily_data = df.resample('D').sum()
```

Step 2: Time Series Analysis

For time series analysis, you can use various techniques like calculating statistics, plotting trends, and identifying seasonality or anomalies. Let's calculate and visualize the daily page views as an example:

```
import matplotlib.pyplot as plt

# Calculate daily page views

daily_page_views = daily_data['page_views']

# Plot daily page views

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

plt.plot(daily_page_views)

plt.title('Daily Page Views Over Time')
plt.xlabel('Date')
plt.ylabel('Page Views')
plt.show()
```

Step 3: User Segmentation

You can segment your users based on their behavior, location, or other attributes. For instance, let's segment users based on their browsing device (mobile vs. desktop):

```
# Create a new column 'device_type' based on user agent data
df['device_type'] = df['user_agent'].str.contains('Mobile', case=False, na=False)
# Segment users
mobile_users = df[df['device_type'] == True]
desktop_users = df[df['device_type'] == False]
```

Step 4: Machine Learning-Based Predictions

To make predictions about website traffic, you can use machine learning models. Let's create a simple example using linear regression to predict future page views based on historical data:

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
# Create features and target variable
X = daily_data[['feature1', 'feature2', ...]]  # Include relevant features
y = daily_data['page_views']
# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Train a linear regression model
model = LinearRegression()
model.fit(X_train, y_train)
# Make predictions
y_pred = model.predict(X_test)
# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse}')
```

This is a basic example, and you can explore more advanced techniques and models depending on your specific website traffic data and the insights you want to gain. Additionally, you can use libraries like scikit-learn for more advanced machine learning tasks.