**Phase 4:**Development part 2

**Title:**Perform the website traffic analysis and create visualization

**INTRODUCTION:**

Analyzing website traffic and building models to gain insights or make predictions can be crucial for understanding user behavior, optimizing content, and making data-driven decisions. Here is a step-by-step guide on how to perform different analyses and model building for website traffic analysis

In website traffic analysis, modeling and evaluation are crucial steps for understanding user behavior, making predictions, and optimizing your website's performance. Here's a guide on how to approach modeling and evaluation in website traffic analysis.

**GIVEN DATASET:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Row | Day | | Day.Of.Week | Date | Page.Loads | Unique.Visits | First.Time.Visits | Returning.Visits | | |
| 1 | Sunday | | 1 | 9/14/2014 | 2,146 | 1,582 | 1,430 | 152 | |  |
| 2 | Monday | | 2 | 9/15/2014 | 3,621 | 2,528 | 2,297 | 231 | |  |
| 3 | Tuesday | | 3 | 9/16/2014 | 3,698 | 2,630 | 2,352 | 278 | |  |
| 4 | Wednesday | | 4 | 9/17/2014 | 3,667 | 2,614 | 2,327 | 287 | |  |
| 5 | Thursday | | 5 | 9/18/2014 | 3,316 | 2,366 | 2,130 | 236 | |  |
| 6 | Friday | | 6 | 9/19/2014 | 2,815 | 1,863 | 1,622 | 241 | |  |
| 7 | Saturday | | 7 | 9/20/2014 | 1,658 | 1,118 | 985 | 133 | |  |
| 8 | Sunday | | 1 | 9/21/2014 | 2,288 | 1,656 | 1,481 | 175 | |  |
| 9 | Monday | | 2 | 9/22/2014 | 3,638 | 2,586 | 2,312 | 274 | |  |
| 10 | Tuesday | | 3 | 9/23/2014 | 4,462 | 3,257 | 2,989 | 268 | |  |
| .. | | …. | | …. | …. | …. | … | ….. | …. |  |
| 1602 | | Friday | | 6 | 2/1/2019 | 4,222 | 2,923 | 2,364 | 559 |  |
| 1603 | | Saturday | | 7 | 2/2/2019 | 2,864 | 1,902 | 1,596 | 306 |  |
| 1604 | | Sunday | | 1 | 2/3/2019 | 3,598 | 2,447 | 2,044 | 403 |  |
| 1605 | | Monday | | 2 | 2/4/2019 | 5,366 | 3,767 | 3,146 | 621 |  |
| 1606 | | Tuesday | | 3 | 2/5/2019 | 5,427 | 3,757 | 3,124 | 633 |  |
| 1607 | | Wednesday | | 4 | 2/6/2019 | 5,667 | 3,759 | 3,130 | 629 |  |
| 1608 | | Thursday | | 5 | 2/7/2019 | 5,517 | 3,763 | 3,069 | 694 |  |
| 1609 | | Friday | | 6 | 2/8/2019 | 4,420 | 2,943 | 2,429 | 514 |  |
| 1610 | | Saturday | | 7 | 2/9/2019 | 3,150 | 2,274 | 1,911 | 363 |  |
| 1611 | | Sunday | | 1 | 2/10/2019 | 4,264 | 2,962 | 2,470 | 492 |  |

****Modeling:****

****Select the Appropriate Model**:**

Choose the modeling techniques that are most suitable for your specific analysis. Common models include time series models, regression models, classification models, clustering models, and machine learning models. The choice depends on the objectives of your analysis.

****Data Preparation****:

Ensure that your dataset is well-prepared for modeling. This includes handling missing values, encoding categorical variables, and scaling/normalizing numerical data. For time series analysis, you'll need to handle the time component appropriately.

****Feature Selection/Engineering****:

Identify and select the relevant features that will be used as inputs to your model. Feature engineering may involve creating new variables or transforming existing ones to enhance model performance.

****Train-Test Split****:

Divide your dataset into a training set and a testing set. The training set is used to train the model, while the testing set is used for evaluation. The common split ratio is 70-80% for training and 20-30% for testing.

****Model Training****:

Train your chosen model on the training data. Ensure that you configure the model's hyperparameters appropriately. For time series data, be mindful of lag values and window sizes.

****Cross-Validation**** **(for machine learning models)**:

Use techniques like k-fold cross-validation to assess model performance on different subsets of your training data. This helps ensure that your model generalizes well to unseen data.

****Model Tuning****:

Fine-tune your model by adjusting hyperparameters to optimize its performance. This may involve grid search or random search for machine learning models.

****Evaluation:****

****Performance Metrics****:

Use relevant performance metrics to evaluate the model's performance. The choice of metrics depends on the type of analysis and the model being used. Common metrics for website traffic analysis include:

* + ****Mean Absolute Error (MAE)**** and ****Mean Squared Error (MSE)**** for regression models.
  + ****Accuracy****, ****Precision****, ****Recall****, and ****F1-score**** for classification models.
  + ****AUC-ROC**** for binary classification models.
  + ****Silhouette Score**** for clustering models.
  + ****Root Mean Squared Error (RMSE)**** for time series forecasting.

****Visualization****:

Create visualizations to help interpret model results. Time series data can be visualized with line plots, while classification results can be visualized with confusion matrices or ROC curves.

****Interpretability****:

For some models, especially machine learning models, it's important to understand the factors that are driving the predictions. Techniques like feature importance analysis or SHAP (SHapley Additive exPlanations) values can help in understanding model decisions.

****Model Comparison**** (**if applicable**):

If you have multiple models, compare their performance to select the best-performing one. This can involve comparing their evaluation metrics and considering factors like model complexity and interpretability.

****Deployment and Monitoring:****

If the model is successful, consider deploying it to make real-time predictions. Continuously monitor the model's performance and update it as needed to ensure it remains accurate over time.

Remember that the choice of models and evaluation metrics should align with the specific goals of your website traffic analysis. Different analyses may require different models and evaluation methods. It's essential to iterate and refine your models to improve their accuracy and usefulness for decision-making.

窗体顶端

窗体底端

# Dataset:

To perform website traffic analysis and create visualizations, you can use publicly available website traffic datasets. One commonly used dataset is the **"Website traffic data set"** from the UCI Machine Learning Repository, which contains hourly website traffic data from various locations. Here's how to access and work with it:

# Step 1:

**Download the Dataset:**

You can download the dataset from the UCI Machine Learning Repository

# Step 2:

**Load the Dataset**

You can use Pandas in Python to load the dataset:

import pandas as pd

data = pd.read\_csv('/home/jovyan/work/path/data-sample.csv')

# Step 3:

**Data Preprocessing:**

Preprocess the data to make it suitable for analysis. This may include handling missing values, data type conversions, and renaming columns. You can find information on data preprocessing in the dataset documentation.

# Step 4:

**Data Analysis and Visualization:**

Now, you can perform analysis and create visualizations using libraries like Matplotlib and Seaborn. Here's an example for creating a time series plot for a specific website traffic analysis:

import matplotlib.pyplot as plt

import seaborn as sns

import pandas as pd

data = pd.read\_csv('/home/jovyan/work/path/data-sample.csv')

fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(18, 10))

fig.suptitle("Website Traffic Data", fontsize=16)

plt.subplots\_adjust(top=0.9)

fig.delaxes(axes[1, 2])

plt.tight\_layout()

Remember to refer to the dataset documentation for information about the features, data format, and any preprocessing required for the specific dataset you choose. Additionally, you can explore government websites or environmental agencies for real-time or historical w e bsite traffic data for specific regions and cities.

# Program:

import pandas as pd

import matplotlib.pyplot as plt import seaborn as sns

data = pd.read\_csv('/home/jovyan/work/path/data-sample.csv')

print(data.head()) # Display the first few rows of the dataset

**Example 1:**

**Subplot:**

import matplotlib.pyplot as plt

import seaborn as sns

import pandas as pd

data = pd.read\_csv('/home/jovyan/work/path/data-sample.csv')

fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(18, 10))

fig.suptitle("Website Traffic Data", fontsize=16)

**Example 2:**

**Histogram**

# Plot First.Time.Visits Distribution

sns.histplot(data['First.Time.Visits'], kde=True, ax=axes[1, 0])

axes[1, 0].set\_title("First Time Visits Distribution")

# Plot Returning.Visits Distribution

sns.histplot(data['Returning.Visits'], kde=True, ax=axes[1, 1])

axes[1, 1].set\_title("Returning Visits Distribution")

**Example 3:**

**Line plot:**

# Plot Day vs. Page Loads

sns.lineplot(x='Day', y='Page.Loads', data=data, ax=axes[0, 0])

axes[0, 0].set\_title("Day vs. Page Loads")

# Plot Date vs. Page Loads

sns.lineplot(x='Date', y='Page.Loads', data=data, ax=axes[0, 2])

axes[0, 2].set\_title("Date vs. Page Loads")

axes[0, 2].tick\_params(axis='x', labelrotation=45)

**Example 4:**

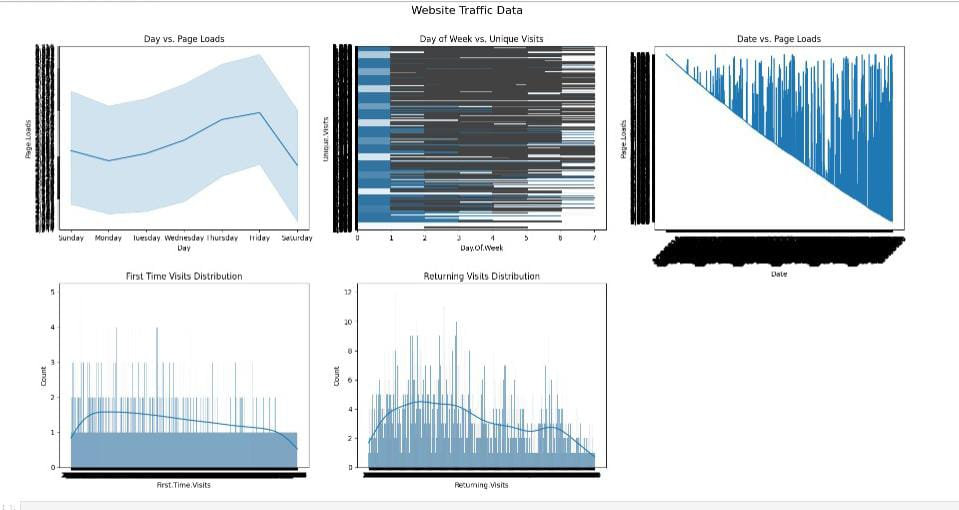
**Barplot:**

# Plot Day.Of.Week vs. Unique.Visits

sns.barplot(x='Day.Of.Week', y='Unique.Visits', data=data, ax=axes[0, 1])

axes[0, 1].set\_title("Day of Week vs. Unique Visits")

# Output:



**Program:**

import matplotlib.pyplot as plt

import seaborn as sns

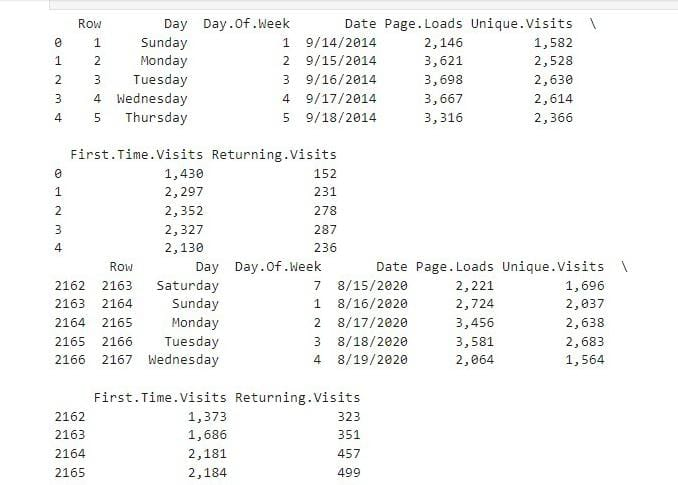
import pandas as pd

data = pd.read\_csv('/home/jovyan/work/path/data-sample.csv')

print(data.head())

print(data.tail())

**Output:**



**Conclusion:**

Analysing website traffic and user behaviour is an indispensable practice for business operating in the digital realm.