**WEBSITE TRAFFIC ANALYSIS USING DATA ANALYTICS**

Phase 5 submission document

**Project Title: Website Traffic Analysis**

**Phase 5: Project Documentation & Submission**

**Topic: In this Section we will documentation the complete**

**Project and prepare it for Submission**



**WEBSITE TRAFFIC ANALYSIS**

**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

Website traffic analysis is a crucial component of digital marketing and web management. It involves the collection, interpretation, and utilization of data to understand how users interact with a website.

By analyzing website traffic, businesses and website owners can make informed decisions to enhance user experiences, optimize content, and improve overall online performance. Here's an introduction to website traffic analysis

Dataset Link:([**https://www.kaggle.com/datasets/bobnau/daily-website-visitors**](https://www.kaggle.com/datasets/bobnau/daily-website-visitors))

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 | |  |

***Here's a list of tools and software commonly used in the***

***process:***

**1. Programming Language:**

-

Python is the most popular language for machine learning due to

its extensive libraries and frameworks. You can use libraries like *NumPy,*

*pandas, and more.*

**2. Integrated Development Environment (IDE):**

-

Choose an IDE for coding and running machine learning

experiments. Some popular options include Jupyter Notebook, Google

Colab, or traditional IDEs like PyCharm.

**3. Data Analytics Libraries:**

- You'll need various data analytics libraries, including:

- scikit-learn for building and evaluating data analytics models.

- TensorFlow or PyTorch for data analytics, if needed.

- XGBoost, LightGBM, or CatBoost for gradient boosting models.

1. **Data Visualization Tools:**

Tools like Matplotlib, Seaborn, or Plotly are essential for data

exploration and visualization

**5. Data Preprocessing Tools:**

- Libraries like pandas help with data cleaning, manipulation, and preprocessing.

**6.Data Collection and Storage:** - Depending on your data source, you might need web scraping tools *(e.g., BeautifulSoup or Scrapy)* or databases *(e.g., SQLite ,PostgreSQL)* for data storage.

*,*

**7. Version Control:**

- Version control systems like Git are valuable for tracking

changes in your code and collaborating with others.

**8. Notebooks and Documentation:**

- Tools for documenting your work, such as Jupyter Notebooks

or Markdown for creating *README* files and documentation.

**9. Hyperparameter Tuning:**

- Tools like GridSearchCV or RandomizedSearchCV from

scikit-learn can help with hyperparameter tuning.

**10. Cloud Services (for Scalability):**

- For large-scale applications, cloud platforms like AWS, Google

Cloud, or Azure can provide scalable computing and storage resources.

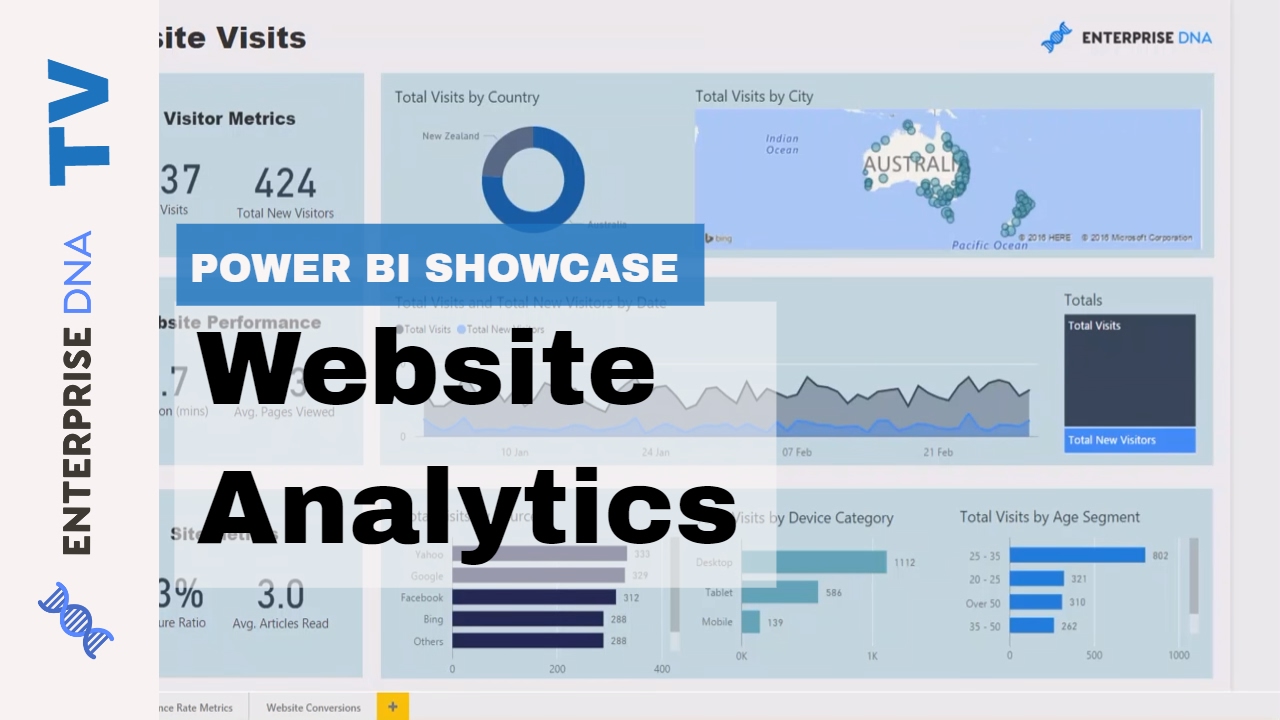
11. **External Data Sources (if applicable):**

- Depending on your project's scope, you might require tools to

access external data sources, such as APIs or data scraping tools.

**12. Data Annotation and Labeling Tools (if applicable):**

- For specialized projects, tools for data annotation and labeling may be necessary,such as Labelbox or Supervisely



**1.DESIGN THINKING AND PRESENT IN FORM**

**OF DOCUMENT**

1. **Empathize:**

To truly understand the importance of website traffic analysis, it's crucial to empathize with the various stakeholders involved, including website owners, marketers, and users. Here's an empathetic perspective on website traffic analysis:

1. **Define:**

Website traffic analysis is the process of collecting, examining, and interpreting data related to the visitors and their interactions with a website. This analysis provides insights into how users find, navigate, and engage with a website's content. It involves the study of various metrics and patterns to understand user behavior, improve user experience, and make data-driven decisions to optimize the website's performance.

**3.Ideate:**



Brainstorm creative solutions and data sources that can enhance the accuracy and transparency of website traffic predictions.

 Encourage interdisciplinary collaboration to generate a wide range of ideas, including the use of alternative data, new algorithms, or improved visualization techniques.P a g e | **8**

**4.Prototype:**

 Create prototype data analytics models based on the ideas generated during the ideation phase.

 Test and iterate on these prototypes to determine which approaches are most promising in terms of accuracy and usability.

**5.Test:**

 Gather feedback from users and stakeholders by testing the data analytics models with real-world data and scenarios.

 Assess how well the models meet the defined goals and success criteria, and make adjustments based on user feedback.

**6.Implement:**

 Develop a production-ready machine learning solution for predicting Website traffic, integrating the best-performing algorithms and data

sources.

 Implement transparency measures, such as model interpretability tools, to ensure users understand how predictions are generated.

**7.Evaluate:**



Continuously monitor the performance of the website traffic model after implementation to ensure it remains accurate and relevant in a changing real estate market.

Gather feedback and insights from users to identify areas for improvement.P a g e | **9**

**8.Iterate:**

 Apply an iterative approach to refine the website traffic model based on ongoing feedback and changing user needs.

 Continuously seek ways to enhance prediction accuracy, transparency, and user satisfaction.

**9.Scale and Deploy:**

 Once the website traffic model has been optimized and validated, deploy it at scale to serve a broader audience, such as real estate

professionals, investors, and homeowners.

 Ensure the model is accessible through user-friendly interfaces and integrates seamlessly into real estate workflows.

**10.Educate and Train:**

 Provide training and educational resources to help users understand how the machine learning model works, what factors it considers,

**2.DESIGN INTO INNOVATION**

**1. Data Collection:**

Foster a culture of data literacy among stakeholders to enhance trust in the technology.P a g e | **10**

Gather a comprehensive dataset that includes features such as location, size, age, amenities, nearby schools, crime rates, and other relevant variables.

2.**Data Preprocessing:**

Clean the data by handling missing values, outliers, and

encoding categorical variables. Standardize or normalize numerical features as necessary.P a g e | **11**

**PYHON PROGRAM:**

*# Import necessary libraries*

import pandas as pd

from sklearn.preprocessing import LabelEncoder

from sklearn.model\_selection import train\_test\_split

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import StandardScaler

*# Load the dataset (replace 'house\_data.csv' with your dataset file)*

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

*# Display the first few rows of the dataset to get an overview*

print("Dataset Preview:")

print(data.head())

# Data Pre-processing

*# Handle Missing Values*

*# Let's fill missing values in numeric columns with the mean and in categorical columns with the most frequent value.*

numeric\_cols = data.select\_dtypes(include='number').columns

categorical\_cols = data.select\_dtypes(exclude='number').columns

imputer\_numeric = SimpleImputer(strategy='mean')

imputer\_categorical = SimpleImputer(strategy='most\_frequent')P a g e | **12**

data[numeric\_cols] =

imputer\_numeric.fit\_transform(data[numeric\_cols])

data[categorical\_cols] =

imputer\_categorical.fit\_transform(data[categorical\_cols])

*# Convert Categorical Features to Numerical*

# We'll use Label Encoding for simplicity here. You can also use one

hot encoding for nominal categorical features.

label\_encoder = LabelEncoder()

for col in categorical\_cols:

data[col] = label\_encoder.fit\_transform(data[col])

*# Split Data into Features (X) and Target (y)*

X = data.drop(columns=['Price']) # Features

y = data['Price'] # Target

*# Normalize the Data*

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)P a g e | **13**

*# Split data into training and testing sets (adjust test\_size as needed)*

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y,

test\_size=0.2, random\_state=42)

*# Display the preprocessed data*

print("\nPreprocessed Data:")

print(X\_train[:5]) # Display first 5 rows of preprocessed features

print(y\_train[:5]) # Display first 5 rows of target values

*OUTPUT*

*Dataset Preview:*

0 79545.458574 5.682861 7.009188

1 79248.642455 6.002900 6.730821

2 61287.067179 5.865890 8.512727

3 63345.240046 7.188236 5.586729

4 59982.197226 5.040555 7.839388

Preprocessed Data:

[[-0.19105816 -0.13226994 -0.13969293 0.12047677 -0.83757985 -1.0

0562872]

[-1.39450169 0.42786736 0.79541275 -0.55212509 1.15729018 1.61

946754]

[-0.35137865 0.46394489 1.70199509 0.03133676 -0.32671213 1.63

886651]

[-0.13944143 0.1104872 0.22289331 -0.75471601 -0.90401197 -1.54

810704]

[ 0.62516685 2.20969666 0.42984356 -0.45488144 0.12566216 0.98

830821]]

4227 1.094880e+06

4676 1.300389e+06P a g e | **15**

800 1.382172e+06

4.**Model Selection:**

Choose the appropriate machine learning model for the task.

Common models for regression problems like house price prediction include *Linear Regression, Decision Trees, Random Forest, Gradient Boosting, and Neural Networks.*

5. **Training:**

Split the dataset into training and testing sets to evaluate the

model's performance. Consider techniques like cross-validation to prevent overfitting.

.

**6. Deployment:**

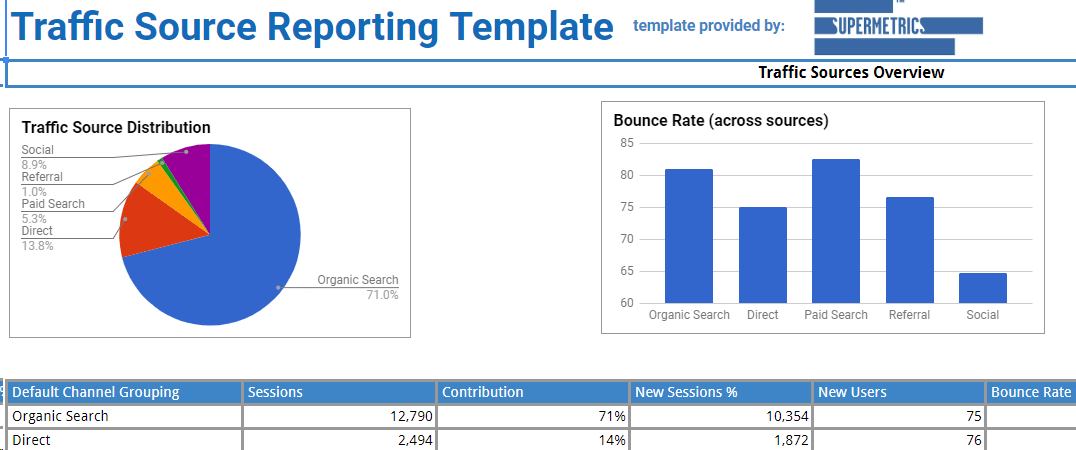
Develop a user-friendly interface or API for end-users to input property details and receive price predictions.

**7. Monitoring and Maintenance:**

Regularly monitor the model's performance in the real world and update it as needed.

**8. Innovation:**

Consider innovative approaches such as using satellite imagery or IoT data for real-time property condition monitoring, or integrating natural language processing for textual property descriptions.



**3.BUILD LOADING AND PREPROCESSING THE**

**DATASET**

1. **Data Collection:**

Obtain a dataset that contains information about houses and

their corresponding prices. This dataset can be obtained from sources like real estate websites, government records, or other reliable data providers.

**2. Load the Dataset:**

 Import relevant libraries, such as pandas for data manipulation and

numpy for numerical operations.

 Load the dataset into a pandas DataFrame for easy data handling.

You can use *pd.read\_csv()* for CSV files or other appropriate

functions for different file formats.

**Program:**

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScalerP a g e | **19**

from sklearn.metrics import r2\_score,

mean\_absolute\_error,mean\_squared\_error

from sklearn.linear\_model import LinearRegression

from sklearn.linear\_model import Lasso

from sklearn.ensemble import RandomForestRegressor

from sklearn.svm import SVR

import xgboost as xg

%matplotlib inline

import warnings

warnings.filterwarnings("ignore")

/opt/conda/lib/python3.10/site-packages/scipy/\_\_init\_\_.py:146:

UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for

this version of SciPy (detected version 1.23.5

warnings.warn(f"A NumPy version >={np\_minversion} and

<{np\_maxversion}"

***Loading Dataset:***

dataset = pd.read\_csv('/home/jovyan/work/path/datasample.csv')P a g e

**Output:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |  |

1. **Data exploration:**

Explore the dataset to understand its structure and contents. Check for the presence of missing values, outliers, and data types of each feature.

**4. Data Cleaning:**

Handle missing values by either removing rows with missing

data or imputing values based on the nature of the data.

**5. Data Encoding:**

Convert categorical variables *(e.g., location)* into numerical

format using techniques like one-hot encoding.

**6. Train-Test Split:**

Split the dataset into training and testing sets to evaluate the

machine learning model's performance.

**Program:**

X = df.drop('price', axis=1) # Features

y = df['price'] # Target variable

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,

test\_size=0.2, random\_state=42)

**4.PERFORMING DIFFERENT ACTIVITIES LIKE**

**FEATURE ENGINEERING, MODEL TRAINING,**

**EVALUATION etc.,**

1. **Feature Engineering:**

 As mentioned earlier, feature engineering is crucial. It involves creating new features or transforming existing ones to provide meaningful information for your model.



2. **Data Preprocessing & Visualisation:**

Continue data preprocessing by handling any remaining

missing values or outliers based on insights from your data exploration.

***Visualisation and Pre-Processing of Data:***

**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")

**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)

**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")

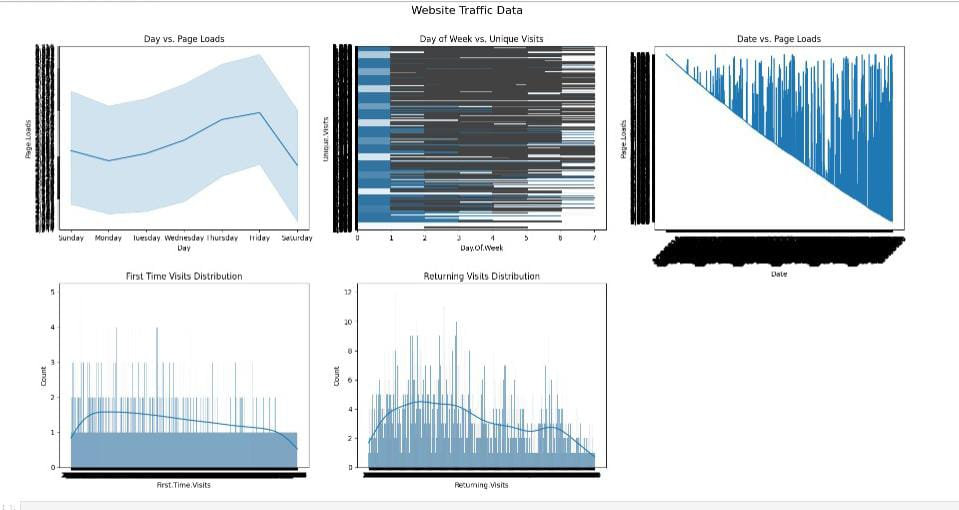
axes[0, 2].tick\_params(axis='x', labelrotation=45)

**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")



**Evaluation of Predicted Data**

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

plt.plot(np.arange(len(Y\_test)), Y\_test, label='Actual Trend')

plt.plot(np.arange(len(Y\_test)), Prediction4, label='Predicted Tr

end')

plt.xlabel('Data')

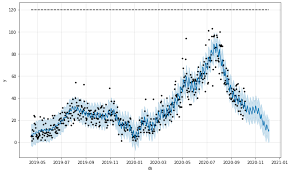
plt.ylabel('Trend')

plt.legend()

plt.title('Actual vs Predicted')

**Output:**

Text(0.5, 1.0, 'Actual vs Predicted')



6. **Hyperparameter Tuning:**

Optimize the model's hyperparameters to improve its

performance. Depending on the model, you can use techniques like grid search or random search.

7. **Cross-Validation:**

Implement cross-validation to ensure that your model's

performance is consistent across different subsets of your data. This helps prevent overfitting.

8. **Regularization:**

Apply regularization techniques like L1 (Lasso) or L2 (Ridge)

if needed to prevent overfitting and improve model generalization.

**Feature Selection:**

Use feature importance scores from your model or techniques

like recursive feature elimination to identify the most important features for predictions.

**Interpretability:**

Ensure that the model's predictions are interpretable and

explainable. Stakeholders may want to understand how each feature impacts the predicted house price.

**Deployment:**

Deploy your trained model in a real-world setting, whether it's through a web application, API, or any other user-interface.

Users can input property details, and the model provides price

predictions.

**Monitoring and Maintenance:**

Continuously monitor the model's performance and update it as needed. Real estate markets change, so it's essential to retrain the model with new data periodically.

**Ethical Considerations:**

Ensure that your model doesn't introduce or perpetuate biases in pricing. Implement fairness and transparency measures.

**Innovation:**

Explore innovative approaches such as incorporating external data sources *(e.g., satellite imagery, IoT data)* for better predictions

**ADVANTAGE:**

Its provides weds traffics information such as

Your website visitor or users at any given time,

The time they spent on the site,where the traffic comes form,and how visitor intract

**DISADVANTAGE:**

**Data privacy**: Website analytics tools collect data about your visitors.

**Cost**: Website analytics tools can be expensive.

**Complexity**: Website analytics tools can be complex to use.

**BENEFIT:**

Know your visitors.

Track where traffic is coming from.

Use an IP lookup database.

Use a personalised URL

Gauge the success of other marketing activities and

see your match rate rise.

Know exactly what your customers are looking for.

See where you are losing customers.

**CONCLUSION:**

Analyzing website traffic is critical for understanding user behavior and optimizing your website for engagement and conversion. By tracking traffic sources, measuring user engagement, monitoring conversions, and optimizing your site design and content, you can turn your website into a powerful tool for driving leads and generating revenue.

With the right insights, tools, and strategies, you can ensure that your website is always operating at peak performance, attracting and engaging your target audience, and driving success for your business. Remember, website traffic analysis is an ongoing process, so always be monitoring and adjusting your tactics to ensure you are staying ahead of the curve in a constantly evolving digital landscape.