WEBSITE TRAFFIC ANALYSIS

PHASE 4 : DEVELOPMENT PART 2

INTRODUCTION

Daily website analysis is the process of examining and evaluating the performance and effectiveness of a website on a day-to-day basis. It is an essential practice for website owners, digital marketers, and web administrators who aim to optimize their online presence, user experience, and achieve specific goals. This ongoing analysis helps in identifying trends, making informed decisions, and implementing improvements.

Website traffic analysis is the process of examining and interpreting the data related to the visitors and their interactions on a website. It is a crucial aspect of web analytics, providing valuable insights into the performance and effectiveness of a website. By analyzing website traffic, businesses and website owners can make informed decisions to optimize their online presence, improve user experience, and achieve their goals

Step 1: Data Visualization

1.1. Histograms

Histograms are a great way to visualize the distribution of individual water quality parameters. They can help us understand the spread and frequency of values within each parameter. For instance, we will create histograms for attributes like pH, hardness, and chlorine concentration.

1.2. Scatter Plots

Scatter plots will allow us to explore the relationships between different water quality parameters. By plotting one parameter against another, we can identify potential correlations or trends. We will create scatter plots for variables like turbidity, sulfate, and organic carbon content.

1.3. Correlation Matrices

Correlation matrices will help us quantify and visualize the relationships between all the variables in our dataset. We will use libraries like Seaborn to generate correlation matrices, providing a clear understanding of how each attribute relates to water potability.

Step 2: Building a Predictive Model

2.1. Data Preprocessing

Before constructing our predictive model, we will perform data preprocessing, which includes handling missing values, feature scaling, and encoding categorical variables if necessary. This step ensures that our data is in a suitable format for modeling.

2.2. Model Selection

We will employ two common machine learning algorithms for classification tasks:

➢ Logistic Regression: This simple yet effective model is suitable for binary classification problems like water potability. We will analyze the relationship between the water quality parameters and the binary outcome variable (potable or nonpotable).

➢ Random Forest: Random Forest is an ensemble learning method that can capture complex relationships in the data. It is robust against overfitting and provides feature importance scores.

2.3. Model Training

We will split our dataset into training and testing sets to train and evaluate our models. This will help us assess their accuracy, precision, recall, and F1 score, among other evaluation metrics.

2.4. Model Evaluation

The evaluation of our models will provide us with insights into how well they can predict water potability. We will compare the performance of both models to select the most suitable one for our analysis.

2.5. Conclusion

Upon completing these steps, we will arrive at a more detailed conclusion about water potability based on the water quality parameters. This conclusion will be supported by data visualizations and our predictive model. We will have a clearer understanding of which parameters are most significant in determining water potability and the accuracy of our predictive model

The Key findings and outcomes from this phase of the project:

- ➢ Insights from data visualizations.
- ➢ Performance and effectiveness of the predictive model.

Outline the next steps in the project, which may include further refining the model, conducting additional analysis, or integrating the model into an application or decision-making process.

PROGRAM

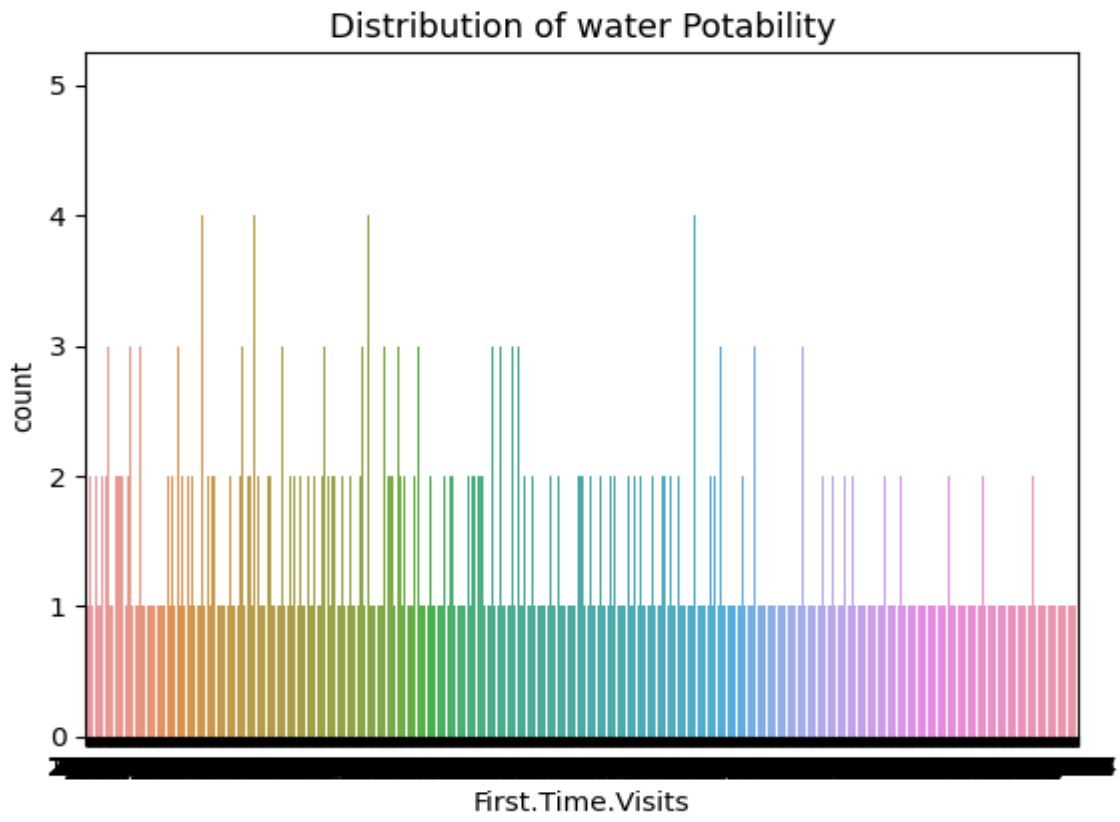
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

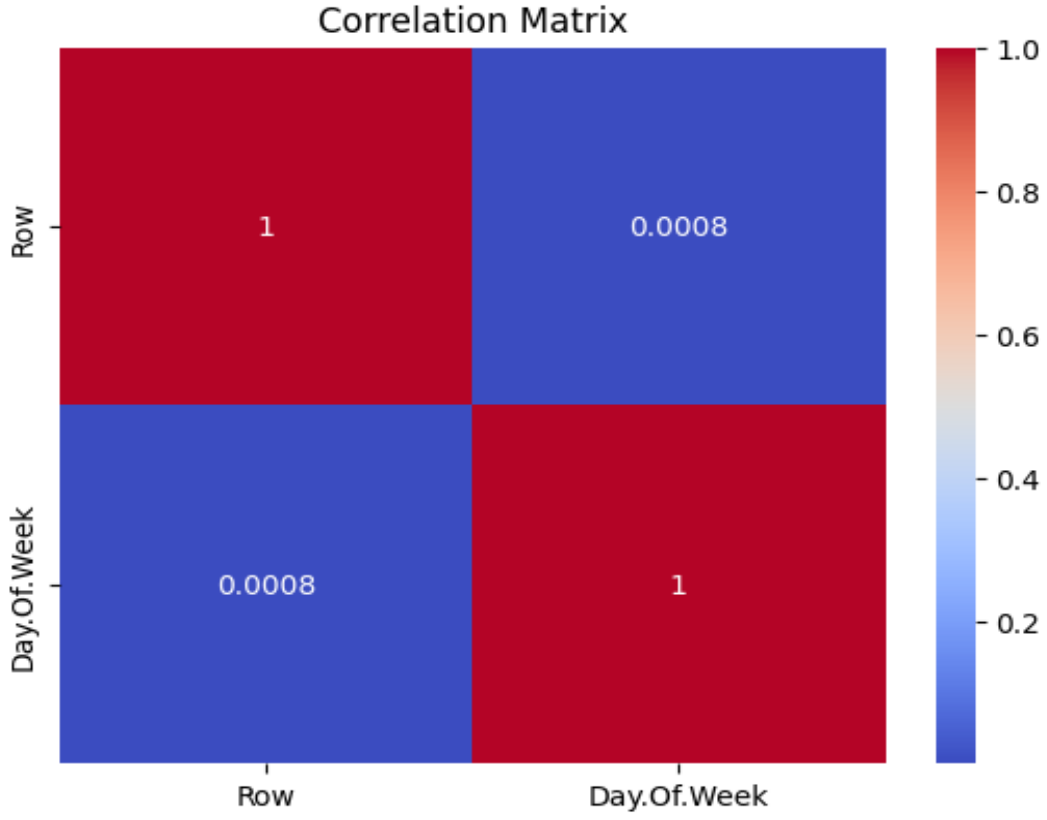
```
# Load the dataset
dataset_url = "https://www.kaggle.com/datasets/bobnau/daily-website-visitors"

# Assuming you've already download the dataset as "daily-website-visitor"
data = pd.read_csv('daily-website-visitors.csv')
data.head()
```

	Row	Day	Day.Of.Week	Date	Page.Loads	Unique.Visits	First.Time.Visits	Returning.Visits
0	1	Sunday	1	9/14/2014	2,146	1,582	1,430	152
1	2	Monday	2	9/15/2014	3,621	2,528	2,297	231
2	3	Tuesday	3	9/16/2014	3,698	2,630	2,352	278
3	4	Wednesday	4	9/17/2014	3,667	2,614	2,327	287
4	5	Thursday	5	9/18/2014	3,316	2,366	2,130	236

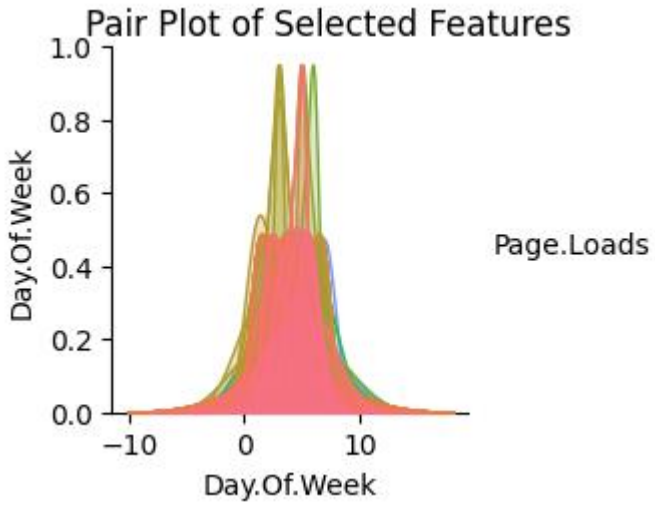
```python
# Display the first few rows of the dataset
print(data.head())

# Summary statistics
print(data.describe())

# Visualize the distribution of potability
sns.countplot(x='First.Time.Visits', data=data)
plt.title('Distribution of First Time Visitors')
plt.show()

# Correlation matrix
correlation_matrix = data.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')

plt.show()
```

```
   Row        Day  Day.Of.Week       Date Page.Loads Unique.Visits  \
0    1     Sunday            1  9/14/2014      2,146         1,582
1    2     Monday            2  9/15/2014      3,621         2,528
2    3    Tuesday            3  9/16/2014      3,698         2,630
3    4  Wednesday            4  9/17/2014      3,667         2,614
4    5   Thursday            5  9/18/2014      3,316         2,366

  First.Time.Visits Returning.Visits
0             1,430              152
1             2,297              231
2             2,352              278
3             2,327              287
4             2,130              236
                Row  Day.Of.Week
count   2167.000000  2167.000000
mean    1084.000000     3.997231
std      625.703338     2.000229
min        1.000000     1.000000
25%      542.500000     2.000000
50%     1084.000000     4.000000
75%     1625.500000     6.000000
max     2167.000000     7.000000
```

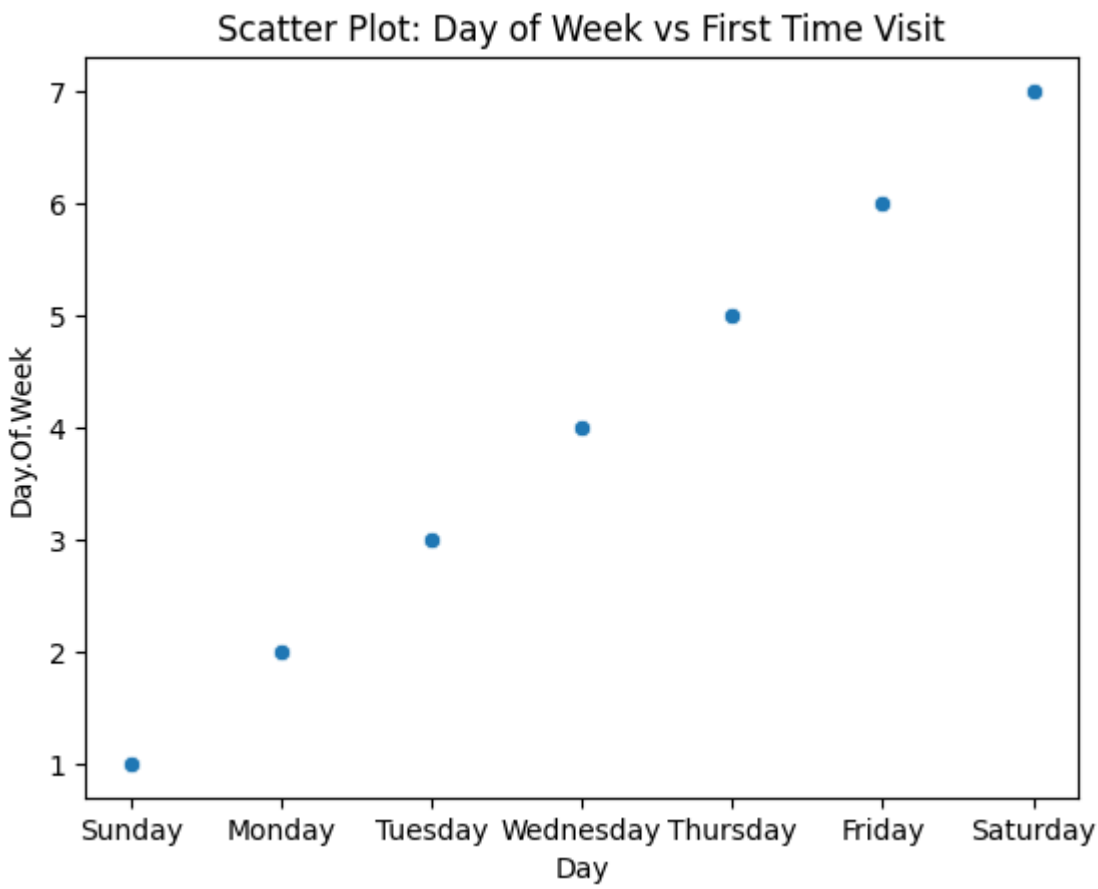

Distribution of water Potability
count
0
1
2
3
4
5
First.Time.Visits


Correlation Matrix
Row
Day.Of.Week
1
0.0008
0.0008
1
Row
Day.Of.Week
1.0
0.8
0.6
0.4
0.2

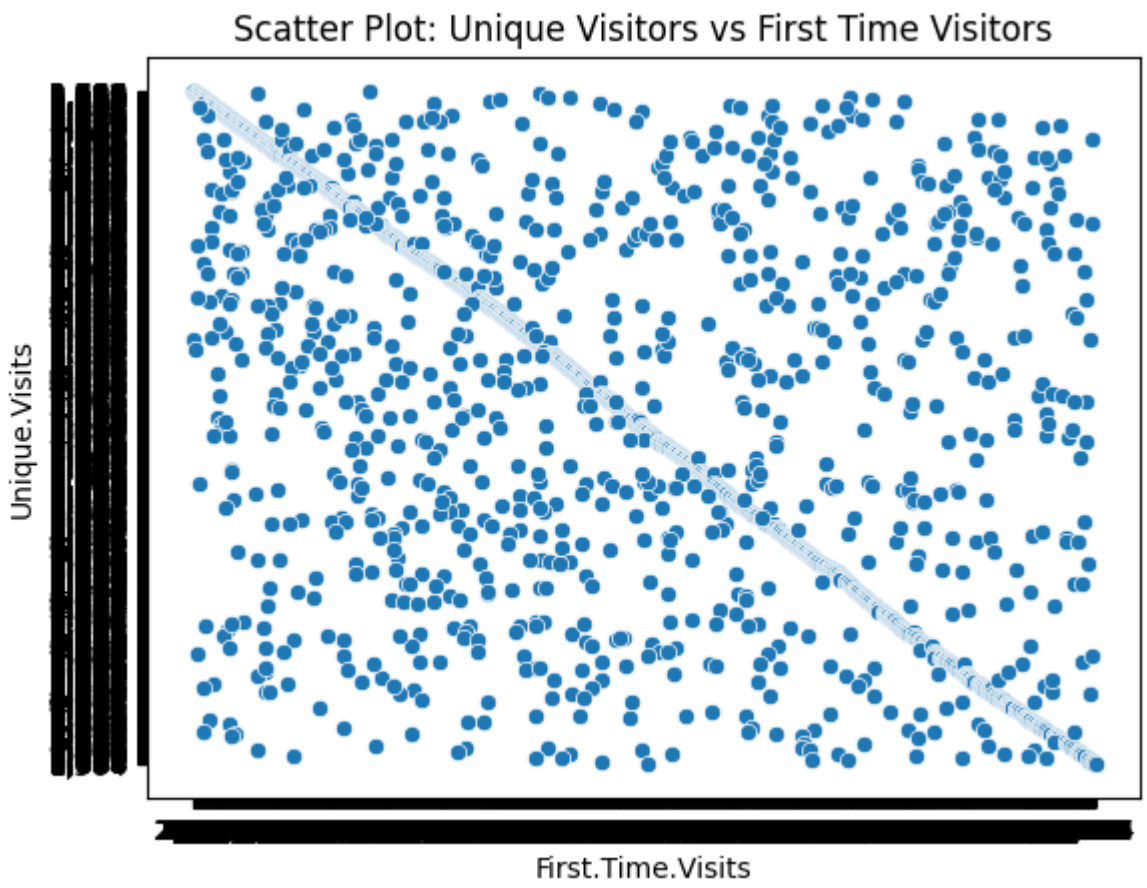
```
# Select a few features for scatter plots
selected_features = ['Day', 'Day.Of.Week', 'Unique.Visits', 'Page.Loads', 'First.Time.Visits']

# Pair plot for selected features
sns.pairplot(data[selected_features], hue='Page.Loads', markers=['o', 's'], palette='husl')
plt.suptitle('Pair Plot of Selected Features', y=1.02)
plt.show()
```




```
#Scatter plot between two specific features
sns.scatterplot(x='Day', y='Day.Of.Week', data=data,palette='husl')
plt.title('Scatter Plot: Day of Week vs First Time Visit')
plt.show()
```




```
# Another example

sns.scatterplot(x='First.Time.Visits', y='Unique.Visits', data=data)
plt.title('Scatter Plot: Unique Visitors vs First Time Visitors')
plt.show()
```


Scatter Plot: Unique Visitors vs First Time Visitors
Unique.Visits
First.Time.Visits

```python
from sklearn.impute import SimpleImputer

# Replace NaN values with the mean of each column
imputer = SimpleImputer(strategy='mean')
x_train = imputer.fit_transform()
x_test = imputer.transform()

# Now, continue with the rest of the code for training the model
rf_classifier = RandomForestClassifier (random_state=42)
rf_classifier.fit(x_train, y_train)

# Make predictions and evaluate the model
y_pred = rf_classifier.predict(x_test)
accuracy = accuracy_score (y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)
print (f'Accuracy: {accuracy}')
print (f'Confusion Matrix: \n{conf_matrix}')
print (f'Classification Report: \n{class_report}')
```

```
Accuracy: 0.6676829268292683
Confusion Matrix:
[[352  60]
 [158  86]]
Classification Report:
              precision    recall  f1-score   support

           0       0.69      0.85      0.76       412
           1       0.59      0.35      0.44       244

    accuracy                           0.67       656
   macro avg       0.64      0.60      0.60       656
weighted avg       0.65      0.67      0.64       656
```

VISUALIZATION USING COGNOS

Dataset Link: https://www.kaggle.com/datasets/bobnau/dailywebsite-visitors

---

Sakthivelan M **25:10:2023**
Nandha College Of Technology