**PREDICTING FUTURE TRAFFIC USING MACHINE LEARNING**

**PHASE 2: DOCUMENTATION**

**PROJECT: WEBSITE TRAFFIC ANALYSIS**



**Introduction:**

* Predicting daily website visitors using a machine learning model can help website owners and marketers make informed decisions about content, marketing strategies, and resource allocation.

**DATA COLLECTION:**

* Collect historical website traffic data, including daily or hourly visitor counts and any relevant features like marketing campaign data, holidays, or special events that could influence traffic.
* Ensure you have a sufficiently large and diverse dataset that covers a representative time period.

**DATA PREPROCESSING:**

* Clean and preprocess the data, handling missing values and outliers.
* Create additional features if needed, such as day of the week, month, seasonality indicators, and lag features (e.g., previous day's traffic).

**DATA SPLITTING:**

* Split the dataset into training, validation, and test sets. Typically, you'd reserve the most recent data for testing.

**DATA SET:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |  |
| 11 | Wednesday | 4 | 9/24/2014 | 4,414 | 3,175 | 2,891 | 284 |  |
| 12 | Thursday | 5 | 9/25/2014 | 4,315 | 3,029 | 2,743 | 286 |  |
| 13 | Friday | 6 | 9/26/2014 | 3,323 | 2,249 | 2,033 | 216 |  |
| 14 | Saturday | 7 | 9/27/2014 | 1,656 | 1,180 | 1,040 | 140 |  |
| 15 | Sunday | 1 | 9/28/2014 | 2,465 | 1,806 | 1,613 | 193 |  |
| 16 | Monday | 2 | 9/29/2014 | 4,096 | 2,873 | 2,577 | 296 |  |
| 17 | Tuesday | 3 | 9/30/2014 | 4,474 | 3,032 | 2,720 | 312 |  |

**FEATURE ENGINEERING:**

* Select relevant features based on domain knowledge and data analysis.
* Normalize or scale numerical features if necessary.

**MODEL SELECTION:**

* Choose a suitable machine learning model for time series forecasting. Common models include ARIMA, SARIMA, Prophet, or more advanced models like LSTM (Long Short-Term Memory) networks.

**MODEL TRAINING:**

* Train the selected model on the training dataset, using the historical traffic data and features.
* Experiment with different hyperparameters and model architectures to optimize performance.

**DATA EVALUATING:**

* Evaluate the model's performance using appropriate time series forecasting metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE).

**PROGRAM:** FUTURE TRAFFIC PREDICTION

**IN [1]:**

import math

from scipy.stats import norm

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from IPython.core.display import HTML

import os

for dirname, \_, filenames in os.walk('/kaggle/input'):

for filename in filenames:

print(os.path.join(dirname, filename))

df = pd.read\_csv("/kaggle/input/daily-website-visitors/daily-website-visitors.csv", \

index\_col = 'Date', thousands = ',', parse\_dates=True)

df.head()

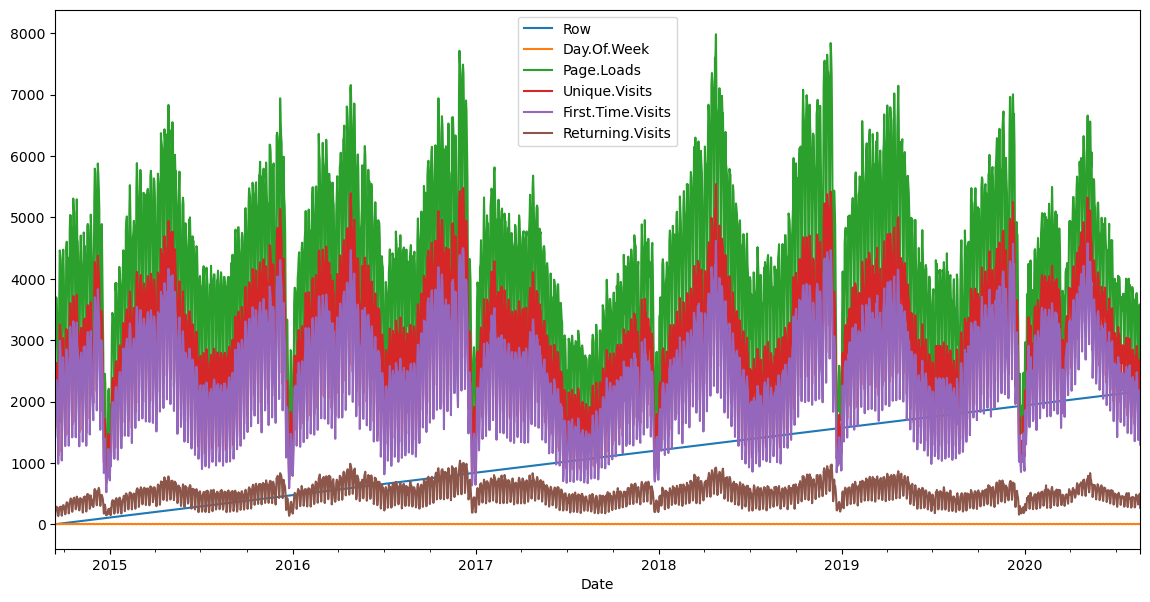
**OUT [1]:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Row | Day | Day.Of.Week | Date | Page.Loads | Unique.Visits | First.Time.Visits |
| 1 | Sunday | 1 | 9/14/2014 | 2,146 | 1,582 | 1,430 |
| 2 | Monday | 2 | 9/15/2014 | 3,621 | 2,528 | 2,297 |
| 3 | Tuesday | 3 | 9/16/2014 | 3,698 | 2,630 | 2,352 |
| 4 | Wednesday | 4 | 9/17/2014 | 3,667 | 2,614 | 2,327 |

**IN [2]:**

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

**OUT [2]:**



**IN [3]:**

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

**OUT [3]:**

mean loads per day: 4116.9893862482695

std deviation of loads per day: 1350.9778426999621

**IN [4]:**

fig, ax1 = plt.subplots()

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

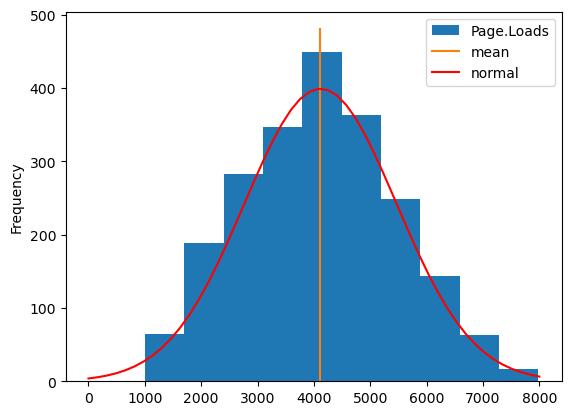
plt.plot([mean, mean], [0, 480], label='mean')

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

plt.legend()

plt.show()

**OUT [4]:**



**IN [5]:**

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

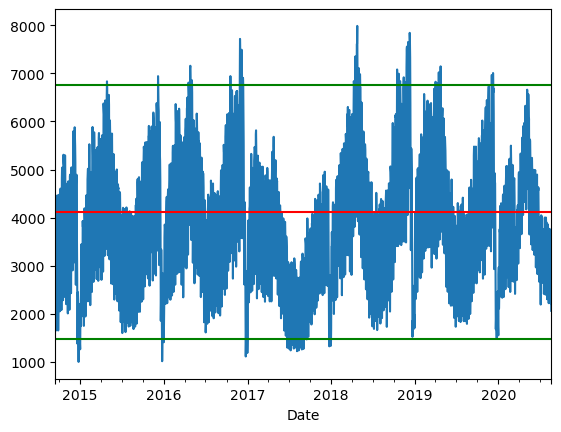
lower = mean - 1.96\*std

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()

**OUT [5]:**



**CONCLUSTION:**

* In conclusion, predicting future traffic trends is a valuable application of machine learning and data analysis that can have significant benefits in various domains, including transportation, urban planning, and website management.
* Predicting future traffic trends is an ongoing process that requires collaboration between data scientists, domain experts, and relevant stakeholders. By following these steps and continuously refining the model, organizations can make informed decisions to optimize traffic management, urban planning, or website strategies. Accurate predictions contribute to improved efficiency, reduced congestion, and better resource allocation, ultimately benefiting both businesses and communities.