EXNO:3 IMPLEMENT PROGRAMS TO CHECK STATIONARY OF A TIME SERIES DATA

DATE:06.02.25

AIM:

Implement programs to check stationary of a time series data

PROCEDURE AND CODE:

Step1:

from google.colab import drive

drive.mount('/content/drive')

Step2:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_absolute\_error

Step3:

file\_path ="/content/daily-website-visitors.csv"

df = pd.read\_csv(file\_path)

print("First 5 rows of the dataset:")

print(df.head())

print("\nColumn names in dataset:", df.columns)

df.columns = df.columns.str.strip()

expected\_columns = ["Date", "Unique.Visits"]

for col in expected\_columns:

if col not in df.columns:

raise ValueError(f"Error: Expected column '{col}' not found. Available columns: {df.columns}")

df['Date'] = pd.to\_datetime(df['Date'])

df.sort\_values('Date', inplace=True)

df['Unique.Visits'] = df['Unique.Visits'].str.replace(',', '').astype(int)

df['Days'] = (df['Date'] - df['Date'].min()).dt.days

X = df[['Days']]

y = df['Unique.Visits']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42, shuffle=False)

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

mae = mean\_absolute\_error(y\_test, y\_pred)

print(f"\nMean Absolute Error (MAE): {mae:.2f}")

plt.figure(figsize=(10, 5))

plt.scatter(df['Days'], df['Unique.Visits'], color='blue', label="Actual Data", alpha=0.5)

plt.plot(X, model.predict(X), color='red', label="Regression Line")

plt.xlabel("Days Since Start")

plt.ylabel("Unique Visitors")

plt.title("Linear Regression Trend")

plt.legend()

plt.show()

future\_days = np.array(range(df['Days'].max() + 1, df['Days'].max() + 31)).reshape(-1, 1)

future\_predictions = model.predict(future\_days)

future\_dates = [df['Date'].max() + pd.Timedelta(days=i) for i in range(1, 31)]

forecast\_df = pd.DataFrame({'Date': future\_dates, 'Predicted\_Visitors': future\_predictions.astype(int)})

print("\nForecasted Visitor Counts for Next 5 Days:")

print(forecast\_df.head())

plt.figure(figsize=(10, 5))

plt.scatter(df['Date'], df['Unique.Visits'], color='blue', label="Actual Data", alpha=0.5)

plt.plot(future\_dates, future\_predictions, color='green', linestyle='dashed', label="Forecasted Data")

plt.xlabel("Date")

plt.ylabel("Unique Visitors")

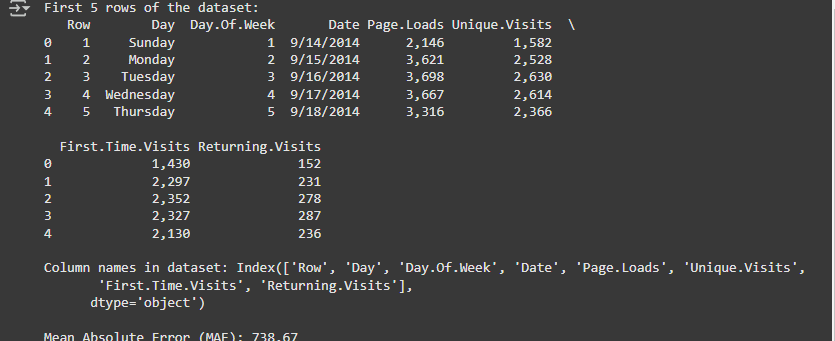
plt.title("Time Series Forecasting")

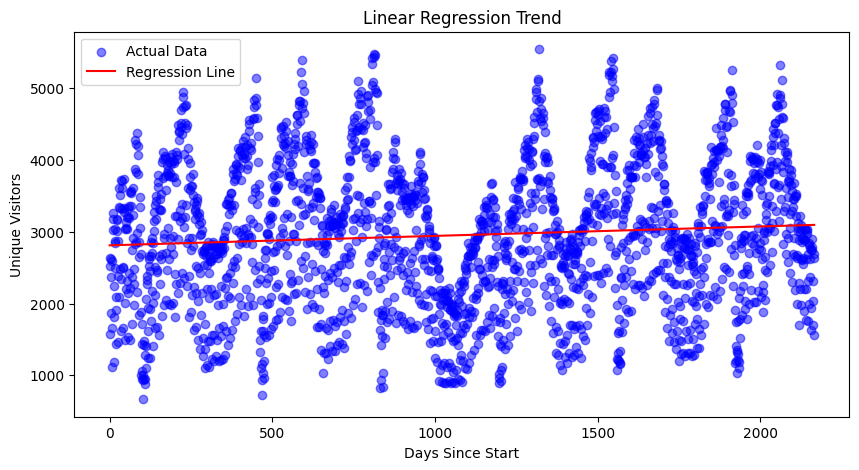
plt.legend()

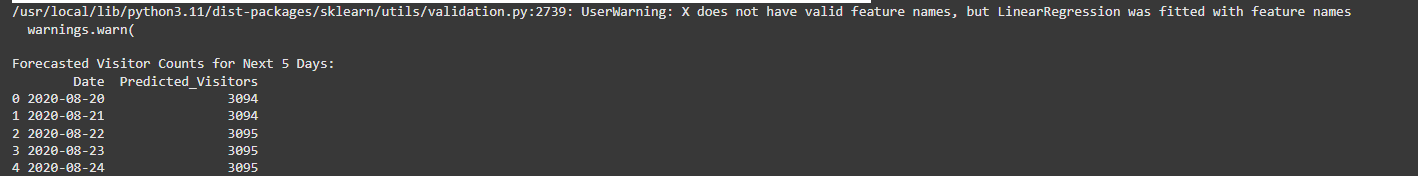
plt.xticks(rotation=45)

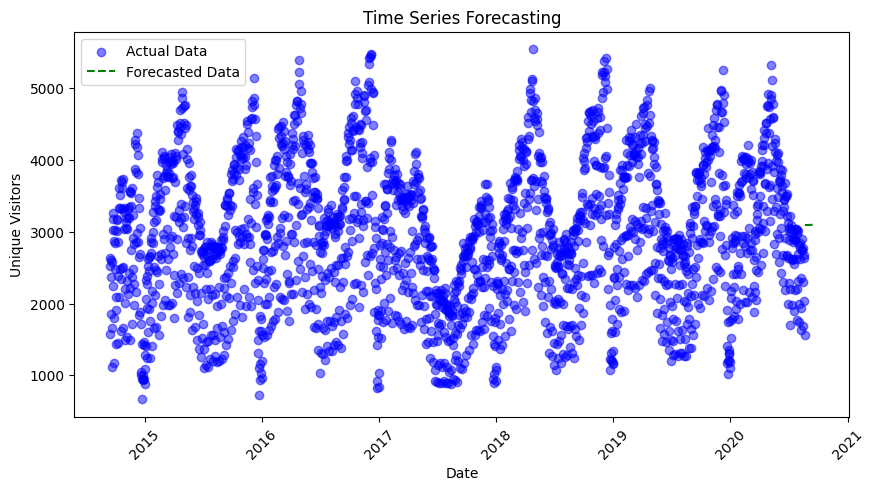
plt.show()

OUTPUTS:









RESULT:  
 The program has been executed successfully.