

EX -4

20/02/2025

Implement programs to check stationary of a time series data

AIM :

Implement programs to check stationary of a time series data

Procedure and Code :

Step 1 - Import the Files and Libraries .

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Step 2 - Describe and Read the Data

```
df=pd.read_csv('/content/drive/MyDrive/TimeSereisDatasets/
daily-website-vvisitors.csv')
```

```
df.head(10)
```

	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

```
df.shape
```

```
(2167, 8)
```

EX -4

20/02/2025

Step 3 - Cleaning and preprocessing the data

```
data_null = df.notnull().sum
```

```
df['Page.Loads'] = df['Page.Loads'].str.replace(',', '').astype(int)
```

```
daywise_data = df.groupby('Day')['Page.Loads'].sum()
```

Step 4 - visualizing the Dataset

```
plt.figure(figsize=(10, 6))
rolmean = data['Page.Loads'].rolling(window=12).mean()
rolstd = data['Page.Loads'].rolling(window=12).std()
plt.plot(data['Page.Loads'], color='blue', label='Original')
plt.plot(rolmean, color='red', label='Rolling Mean')
plt.plot(rolstd, color='black', label='Rolling Std')
plt.legend(loc='best')
plt.title('Rolling Mean & Standard Deviation of Page Loads')
plt.show()
```

Step -5 : Result

```
result = adfuller(data['Page.Loads'].dropna())
print('ADF Statistic:', result[0])
print('p-value:', result[1])
print('Critical Values:')
```

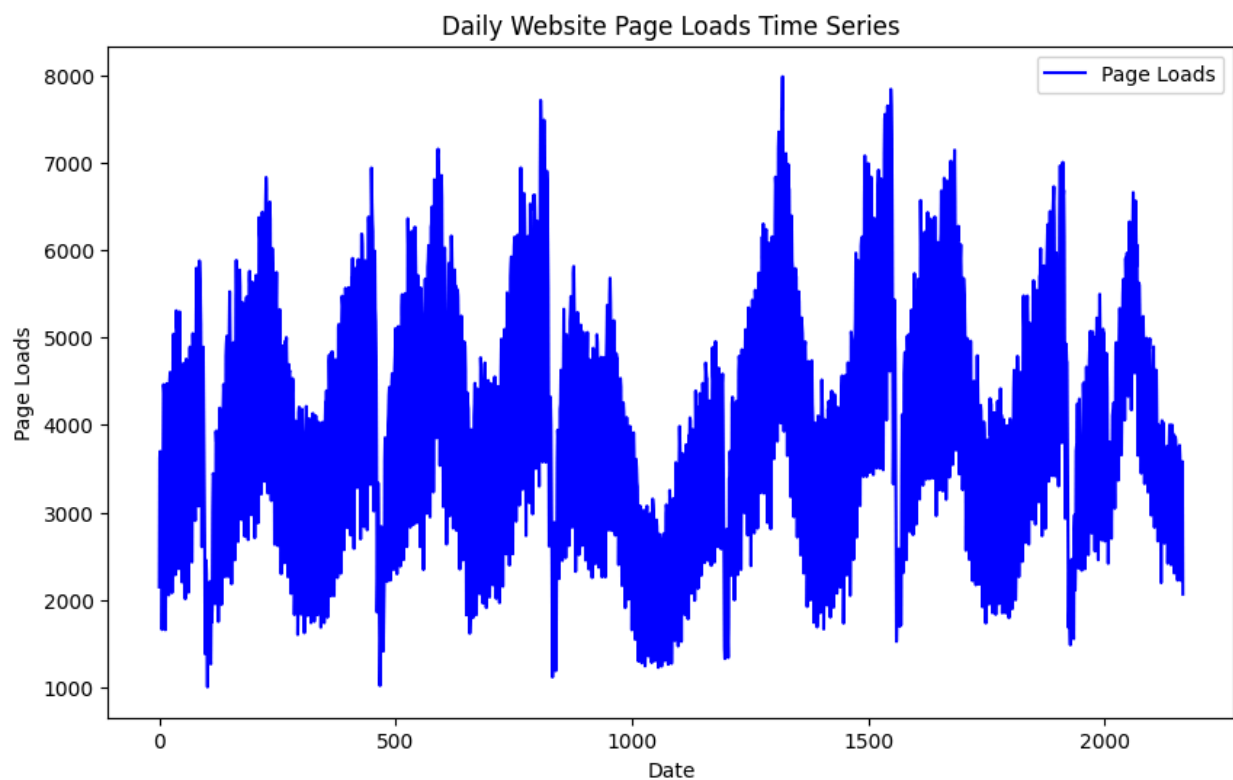
EX -4

20/02/2025

```
for key, value in result[4].items():  
    print(f'    {key}: {value}')
```



```
if result[1] < 0.05:  
    print("The time series is stationary (reject the null hypothesis).")  
else:  
    print("The time series is not stationary (fail to reject the null  
hypothesis).")
```



EX -4
20/02/2025

Result:

Thus the Program has been Executed Successfully.