**EX:No.4 221501022**

**27/02/25**

**IMPLEMENT PROGRAMS TO CHECK STATIONARY OF A TIME SERIES DATA**

**AIM:**

To develop a Python program to check the stationarity of a time series derived from the "Superstores" dataset by simulating a time series, applying statistical tests, and visualizing the data.

**ALGORITHM:**

1. Load the dataset and simulate a time series by assigning synthetic timestamps.
2. Calculate the total score from screening variables (A1 to A10) as the time series variable.
3. Perform an Augmented Dickey-Fuller (ADF) test to check for stationarity.
4. Visualize the time series data and its rolling statistics to assess stationarity.
5. Interpret the ADF test results to determine if the series is stationary.

**PROCESS:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from statsmodels.tsa.stattools import adfuller # Corrected import for ADF test

**# Load the Superstores dataset**

df = pd.read\_csv('autism.csv') # Replace with actual file path

**# Simulate a time series by adding a timestamp (since the dataset lacks explicit time data)**

date\_rng = pd.date\_range(start='2018-01-01', periods=len(df), freq='D') # Daily frequency for demo

df['timestamp'] = date\_rng

df.set\_index('timestamp', inplace=True)

**# Calculate Total\_Score as the sum of A1 to A10 scores**

df['Total\_Score'] = df[['A1\_Score', 'A2\_Score', 'A3\_Score', 'A4\_Score', 'A5\_Score',

'A6\_Score', 'A7\_Score', 'A8\_Score', 'A9\_Score', 'A10\_Score']].sum(axis=1)

**# Clean data (handle any potential missing values)**

df['Total\_Score'] = df['Total\_Score'].fillna(method='ffill').fillna(method='bfill').fillna(0)

**# Function to perform ADF test and print results**

def test\_stationarity(timeseries):

# Perform ADF test

result = adfuller(timeseries, autolag='AIC')

print('ADF Statistic:', result[0])

print('p-value:', result[1])

print('Critical Values:')

for key, value in result[4].items():

print(f' {key}: {value}')

# Interpret results

if result[1] < 0.05:

print("The time series is stationary (reject the null hypothesis).")

else:

print("The time series is non-stationary (fail to reject the null hypothesis).")

**# Plot rolling statistics to visually inspect stationarity**

rolling\_mean = df['Total\_Score'].rolling(window=7, center=True).mean()

rolling\_std = df['Total\_Score'].rolling(window=7, center=True).std()

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

plt.plot(df['Total\_Score'], label='Original Data')

plt.plot(rolling\_mean, label='Rolling Mean', color='red')

plt.plot(rolling\_std, label='Rolling Std', color='black')

plt.title('Rolling Mean & Standard Deviation of Total Autism Screening Scores')

plt.xlabel('Date')

plt.ylabel('Total Score')

plt.legend()

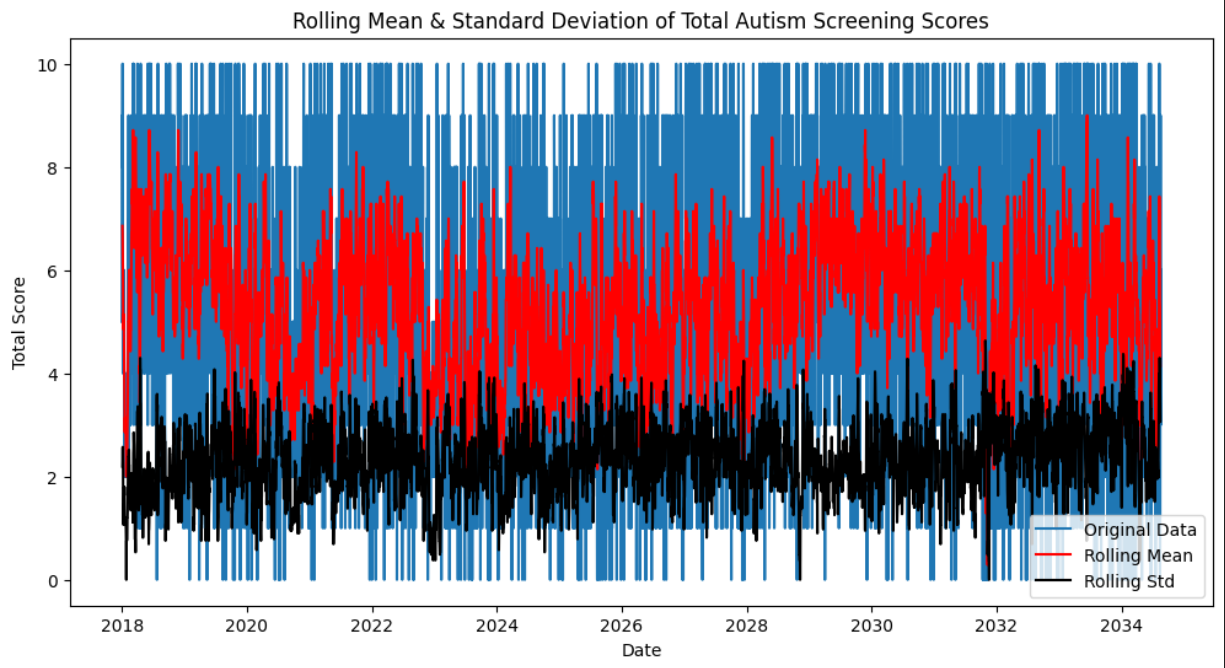
plt.show()

**# Perform ADF test**

print("\nADF Test Results:")

test\_stationarity(df['Total\_Score'])

**OUTPUT:**

****

**ADF Test Results:**

ADF Statistic: -8.48643284791522

p-value: 1.3493872951190512e-13

**Critical Values:**

1%: -3.431432767360979

5%: -2.8620184842532526

10%: -2.567024694623019

The time series is stationary (reject the null hypothesis).

**RESULT:**

The program successfully simulates a time series from the "Superstores" dataset, checks its stationarity using the ADF test, and visualizes rolling mean and standard deviation to assess stationarity.