**Ex.no: 04 221501048**

**07/03/2025**

**Checking stationarity of time series data using augmented dickey fuller test**

## AIM:

To check stationarity of time series data using augmented dickey fuller test.

**PROCEDURE:**

# Import Required Libraries

* + pandas: For data manipulation and loading the dataset.
  + statsmodels.tsa.stattools.adfuller: For performing the Augmented Dickey-Fuller test.
  + matplotlib.pyplot: For visualizing the time series data.

# Load the Dataset

* + Read the CSV file containing time series data (Gold\_Price\_DataSet.csv).
  + Set the 'Date' column as the index.
  + Handle errors such as:
    - Missing file.
    - Incorrect column name.

# Extract the Time Series Data

* + Select the 'Price' column (assumed to contain time series values).

# Perform the Augmented Dickey-Fuller (ADF) Test

* + Apply the adfuller() function to check stationarity.
  + Print the results:
    - ADF Statistic
    - p-value
    - Critical values at 1%, 5%, and 10% confidence levels.

# Interpret the Results

* + If p-value <= 0.05: The time series is **stationary**.
  + Else: The time series is **non-stationary**.

# Plot the Time Series Data

* + Visualize the 'Price' column against 'Date' to observe trends and patterns.

**Code:**

import pandas as pd

from statsmodels.tsa.stattools import adfuller import matplotlib.pyplot as plt

try:

data = pd.read\_csv('/content/Gold\_Price\_DataSet.csv', index\_col='Date') # Assuming 'Date' is your date/time index column

except FileNotFoundError:

print("Error: '/content/Gold\_Price\_DataSet.csv' not found. Please upload your data file.")

data = None except KeyError:

print("Error: 'Date' column not found in the CSV. Please specify correct index column name.")

data = None

if data is not None:

# Extract the time series data

# The column name was changed from 'value' to 'Price'

timeseries = data['Price'] # Assuming 'Price' is your time series data column

# Perform the Augmented Dickey-Fuller test result = adfuller(timeseries)

# Print the test results

print('ADF Statistic: %f' % result[0]) print('p-value: %f' % result[1]) print('Critical Values:')

for key, value in result[4].items(): print('\t%s: %.3f' % (key, value))

print(result[1])

# Interpret the results if result[1] <= 0.05:

print("The time series is likely stationary.") else:

print("The time series is likely non-stationary.")

## OUTPUT:

ADF Statistic: -3.140151

p-value: 0.023721 Critical Values:

1%: -3.433

5%: -2.863

10%: -2.567

0.02372101223380104

The time series is likely stationary.

plt.figure(figsize=(12, 6)) plt.plot(data.index, timeseries) plt.title('Gold Price Time Series') plt.xlabel('Date') plt.ylabel('Price')

plt.grid(True) plt.show()

## OUTPUT:

****

**RESULT:**

The program for checking a time series data stationary or not has been successfully implemented .