EX:No.3 221501042

**Implement programs to check stationary of a time series data**

**Aim:**

Write a program to check stationary of a time series data

**Algorithm:**

1.**Import Libraries**

* Load required Python libraries like pandas, matplotlib, and statsmodels.

2.**Load Dataset**

* Read the weather dataset (weather\_data.csv) and set the **'Date'** column as the index.

3. **Select a Time Series Column**

* Choose a column for analysis .

4.**Visualize the Data**

* Plot the **original time series**.
* Compute and plot the **rolling mean** and **rolling standard deviation** to observe trends.

5.**Apply the Augmented Dickey-Fuller (ADF) Test**

* If **p-value < 0.05**, the data is stationary.
* Else, the data is non-stationary.

6.**Apply the KPSS Test**

* If **p-value > 0.05**, the data is stationary.
* Else, the data is non-stationary.

7.**Display the Results**

* Print the conclusions from both tests.
* If results are conflicting, further transformations may be needed.

**Code:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import statsmodels.api as sm

from statsmodels.tsa.stattools import adfuller, kpss

# Load the weather dataset

df = pd.read\_csv(r"C:\Users\harsh\Downloads\cleaned\_weather.csv", parse\_dates=['date'], index\_col='date')

time\_series = df['VPmax']

# Function to perform Augmented Dickey-Fuller Test

def adf\_test(series):

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

print("Augmented Dickey-Fuller Test:")

print(f"Test Statistic: {result[0]}")

print(f"p-value: {result[1]}")

print(f"Critical Values: {result[4]}")

if result[1] < 0.05:

print("The data is stationary (Reject null hypothesis).")

else:

print("The data is non-stationary (Fail to reject null hypothesis).")

print("-" \* 50)

# Function to perform KPSS Test

def kpss\_test(series):

result = kpss(series, regression='c', nlags='auto')

print("KPSS Test:")

print(f"Test Statistic: {result[0]}")

print(f"p-value: {result[1]}")

print(f"Critical Values: {result[3]}")

if result[1] > 0.05:

print("The data is stationary (Fail to reject null hypothesis).")

else:

print("The data is non-stationary (Reject null hypothesis).")

print("-" \* 50)

# Plot the time series with rolling mean and standard deviation

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

plt.plot(time\_series, label='Original Time Series', color='blue')

plt.plot(time\_series.rolling(window=12).mean(), label='Rolling Mean', color='red')

plt.plot(time\_series.rolling(window=12).std(), label='Rolling Std Dev', color='green')

plt.title('Vapour Pressure Time Series with Rolling Statistics')

plt.legend()

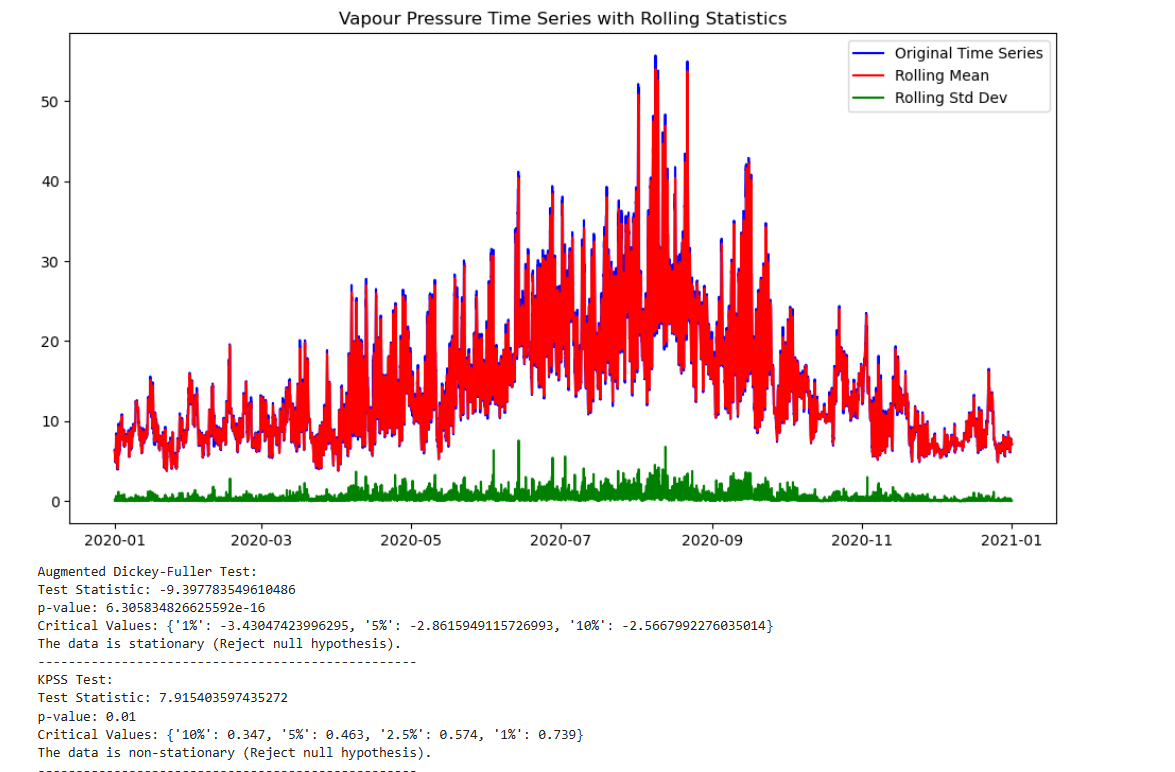
plt.show()

# Apply Stationarity Tests

adf\_test(time\_series)

kpss\_test(time\_series)

**Output:**



**Result:**

Thus, the program to check the stationarity of time series data was checked successfully.