**EX:No.4 221501013**

**04/03/25**

**IMPLEMENT AN PYTHON PROGRAM TO CHECK STATIONARY OF TIME SERIES DATA**

**AIM:**

To implement linear regression model using time series dataset.

**PROCESS:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from statsmodels.tsa.stattools import adfuller

**# Load the dataset**

file\_path = r'Electric\_Production.csv'

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

**# Check the columns to see what data is available**

print("Columns in the dataset:", data.columns)

**# Ensure there are no leading/trailing spaces in column names**

data.columns = data.columns.str.strip()

**# Check if 'IPG2211A2N' column exists**

if 'IPG2211A2N' not in data.columns:

raise KeyError("'IPG2211A2N' column not found in the dataset. Please check the column name.")

**# Access the 'IPG2211A2N' column**

production\_data = data['IPG2211A2N']

**# Reverse the order of the data to maintain chronological order**

production\_data\_reverse = production\_data.iloc[::-1]

**# Reset index to maintain the correct time series order**

production\_data\_reverse.reset\_index(drop=True, inplace=True)

**# Handle Missing Values**

data['IPG2211A2N'].fillna(data['IPG2211A2N'].mean(), inplace=True)

**# Handle Outliers using IQR for 'IPG2211A2N' column**

Q1 = data['IPG2211A2N'].quantile(0.25)

Q3 = data['IPG2211A2N'].quantile(0.75)

IQR = Q3 - Q1

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

**# Filter data to remove outliers based on the IQR bounds**

data = data[(data['IPG2211A2N'] >= lower\_bound) & (data['IPG2211A2N'] <= upper\_bound)]

**# Plot the Time Series to visually inspect**

plt.figure(figsize=(10, 6))

plt.plot(production\_data\_reverse)

plt.title('Electric Production Data (Time Series Plot)')

plt.xlabel('Time')

plt.ylabel('Electric Production')

plt.grid(True)

plt.show()

**# Perform Augmented Dickey-Fuller (ADF) Test for Stationarity**

result = adfuller(production\_data\_reverse)

# Extract and print the ADF test results

print("ADF Test Results:")

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

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

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

**# Interpretation of the ADF test**

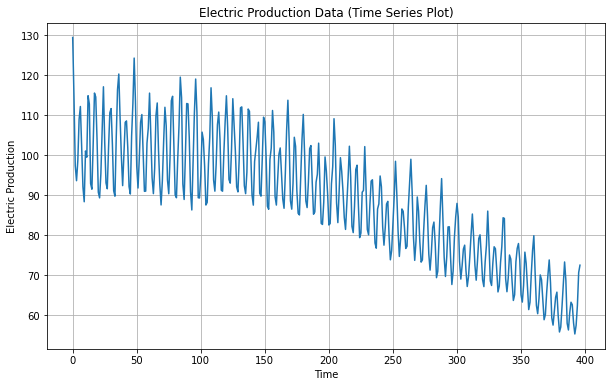
if result[1] <= 0.05:

print("The series is likely stationary (p-value <= 0.05).")

else:

print("The series is likely non-stationary (p-value > 0.05).")

**OUTPUT:**

****

**RESULT:**

The program to implement of program to check stationary or not is created and executed successfully.