No.5 221501015

22/04/25

IMPLEMENT PROGRAMS TO CHECK STATIONARY OF BITCOIN PRICE DATA

AIM:

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

ALGORITHM:

Load the Bitcoin price dataset, ensuring a 'Date' column that can be set as the index.

Select the 'Close' price column as the time series variable for analysis.

Handle any missing values in the 'Close' price data to ensure data integrity.

Perform an Augmented Dickey-Fuller (ADF) test on the 'Close' price time series to check for stationarity.

Visualize the Bitcoin 'Close' price data and its rolling statistics (mean and standard deviation) to visually assess stationarity.

Interpret the ADF test results (ADF statistic and p-value) in relation to critical values to determine if the Bitcoin price 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 Bitcoin price data

df = pd.read\_csv('BTC-USD.csv') # Replace with actual file path to your Bitcoin CSV

df['Date'] = pd.to\_datetime(df['Date'])

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

# Select the 'Close' price as our time series

df = df[['Close']].copy() # Focus on the 'Close' price

# Clean data (handle any potential missing values)

df['Close'] = df['Close'].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['Close'].rolling(window=7, center=True).mean()

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

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

plt.plot(df['Close'], label='Original Data (Close Price)')

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 Bitcoin Close Price')

plt.xlabel('Date')

plt.ylabel('Close Price (USD)')

plt.legend()

plt.show()

# Perform ADF test

print("\nADF Test Results:")

test\_stationarity(df['Close'])

OUTPUT:

ADF Test Results:

ADF Statistic: -1.503648827973287

p-value: 0.5324874018376769

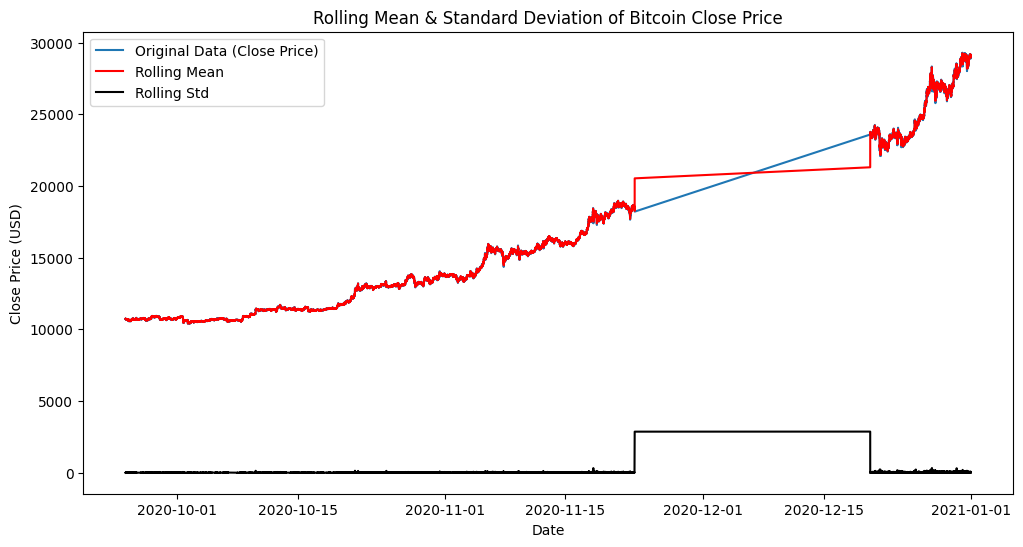
Critical Values:

1%: -3.436899774193266

5%: -2.864489231808213

10%: -2.5683473353431237

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



RESULT:

The program successfully loads the Bitcoin price dataset, checks the stationarity of the 'Close' price time series using the ADF test, and visualizes the rolling mean and standard deviation to aid in assessing stationarity. The ADF test indicates that the Bitcoin 'Close' price time series is non-stationary based on the provided data.