



**UNIVERSITY OF ECONOMICS AND LAW**

Faculty of Finance and Banking

**CRO'S PRICE PREDICTION THROUGH ARIMA**  
-Crypto.com Chain-

# **Python Package for Finance**

## **REPORT**

**Võ Thụy Uyên Nhi**

**K194141736**

**Lector: Dr. Ngô Phú Thanh**

## Mục lục:

<b>1/ Giới thiệu đề tài:</b>	<b>2</b>
1.1/ Crypto.com Chain (CRO)	3
1.2/ Mô hình ARIMA	3
<b>2/ Dữ liệu và phương pháp thực hiện:</b>	<b>3</b>
<b>3/ Code:</b>	<b>5</b>
3.1 Thư viện:	5
3.1.1 Cài đặt các gói thư viện:	5
3.1.2 Khai báo thư viện:	5
3.1.3 Viết các hàm cần dùng:	5
3.2 Thu thập dữ liệu:	6
3.2.1 Tải dữ liệu lên, làm sạch dữ liệu và định dạng cho dữ liệu	6
3.2.2 Khảo sát chuỗi thời gian và kiểm tra tính ổn định của dữ liệu	7
3.3 Xây dựng mô hình	11
3.3.1 Lựa chọn tham số ARIMA(p, d, q)	11
3.3.2 Chạy mô hình và phân tích kết quả mô hình ARIMA	13
3.3.3 Phương pháp Auto ARIMA	14
3.3.4 Hồi quy mô hình SARIMA	14
3.3.5 Kết quả dự báo và thảo luận:	15
<b>Tham khảo:</b>	<b>17</b>

## 1/ Giới thiệu đề tài

Trong những năm gần đây, tiền mã hóa hay còn gọi là cryptocurrency đã và đang phát triển một cách nhanh chóng và ngày càng phổ biến trên thế giới. Tuy nhiên, đây là một lĩnh vực vẫn còn khá mới, có nhiều rủi ro, chủ yếu được đặc trưng bởi sự biến động về giá và số lượng giao dịch của chúng. Vì thế, em muốn chọn đây là một đề tài để nghiên cứu để dự đoán những biến động về giá của đồng coin thông qua phương pháp học máy, mô hình chuỗi thời gian ARIMA

Trong bài báo cáo này em sẽ lựa chọn Crypto.com Chain để làm đối tượng nghiên cứu

### 1.1/ Crypto.com Chain (CRO)

Crypto.com Chain (mã: CRO) là đồng tiền mã hoá trong nền tảng nền tảng Blockchain Crypto.com.

Crypto.com Chain là một Blockchain công khai, mã nguồn mở, phi tập trung với tốc độ cao và phí thấp, được thiết kế để trở thành một Blockchain giúp thúc đẩy việc áp dụng hàng loạt công nghệ thông qua các trường hợp sử dụng như Payments, DeFi và NFT

Chọn đồng Cronos bởi vì Cronos coin được thành lập bởi Crypto.com là một trong những sàn giao dịch có vốn hóa lớn nhất Crypto, giúp người dùng có thể mua bán tài sản tiền điện tử, tương tự các sàn CEX khác như Binance, Okex, Huobi,... Hiện tại, Crypto.com đang nằm trong top 10 Volume giao dịch trong 24h theo CoinGecko. Và bởi vì các thành viên đội dev đều có kinh nghiệm nhiều năm trong tài chính. Nên em kì vọng nó có thể sẽ bùng nổ trong tương lai.

Hiện tại, Crypto.com Chain (CRO) đang được chạy trên nền tảng của Ethereum theo tiêu chuẩn ERC-20.

### 1.2/ Mô hình ARIMA

Mô hình ARIMA - Autoregressive Integrated Moving Average là một trong những loại mô hình chung nhất để dự báo một chuỗi thời gian có thể được tạo ra là cố định bằng cách phân tích, kết hợp với các phép biến đổi phi tuyến như ghi nhật ký hoặc xác định.

Mô hình sử dụng cho việc dự báo giá là ARIMA, bởi nó có nhiều lợi ích phù hợp với đề tài. Mô hình chuỗi thời gian cung cấp cơ hội để dự báo các giá trị trong tương lai. Dựa trên các giá trị trước đó, giá trị trong quá khứ mà chuỗi thời gian có thể sử dụng được để dự báo các xu hướng về kinh tế, thời tiết và lập kế hoạch năng lực, để nêu tên

một số. Các thuộc tính cụ thể của dữ liệu chuỗi thời gian có nghĩa là các phương pháp thống kê chuyên biệt thường được yêu cầu.

Mô hình ARIMA hướng tới việc đưa ra các dự báo đáng tin cậy về chuỗi thời gian thông qua các khái niệm về tự tương quan, tính ổn định và tính thời vụ.

## **2/ Dữ liệu và phương pháp thực hiện**

Với mục đích đề tài của mình, em sẽ sử dụng ARIMA model chuỗi thời gian không có xu hướng và tính thời vụ làm kỹ thuật dự đoán để dự đoán giá đóng cửa của Cronos.

Tính thời vụ là một loại mô hình tự tương quan nhất định xảy ra lặp đi lặp lại sau những khoảng thời gian đều đặn, ví dụ hàng tháng, hàng quý. Thời vụ phải được sửa chữa trước khi đào tạo mô hình vì mô hình có thể xem xét các kiểu thời vụ khác nhau sẽ cho kết quả sai. Xu hướng là sự tăng hoặc giảm dài hạn trong mức độ của chuỗi thời gian. Nếu các xu hướng không được phân tích, chúng sẽ dẫn đến việc đọc sai và diễn giải sai tập dữ liệu.

Mô hình ARIMA theo mùa được ký hiệu là ARIMA (p, d, q) (P, D, Q) m với P, D và Q là các phần tự hồi quy theo mùa, phân biệt và trung bình động và m là số giai đoạn trong mỗi mùa. trong khi mô hình ARIMA không theo mùa được phân loại là mô hình ARIMA (p, d, q), trong đó:

ARIMA:

- AR (autoregressive) model: p là số điều khoản tự hồi quy
- I (integration): d là số chênh lệch cần thiết cho tính ổn định
- MA (moving average) model: q là số lỗi dự báo trễ trong phương trình dự đoán (tức là thứ tự của mô hình trung bình động)

Data được lấy từ cổng dữ liệu Yahoo Finance bởi sự thuận tiện và chính xác của nó.

Quy trình thực hiện:

- Thu tập dữ liệu:
  - Cài đặt và khai báo thư viện cần dùng
  - Tải dữ liệu lên, làm sạch dữ liệu và định dạng cho dữ liệu
  - Khảo sát chuỗi thời gian và kiểm tra tính ổn định của dữ liệu
  - Kiểm tra yếu tố mùa vụ
- Xây dựng mô hình:
  - Lựa chọn tham số ARIMA(p, d, q)

Chạy mô hình và phân tích kết quả mô hình ARIMA

Phương pháp Auto ARIMA

Hồi qui mô hình SARIMA

Dự báo

### **3/ Code**

#### **3.1 Thư viện**

##### **3.1.1 Cài đặt các gói thư viện**

```
!pip install yfinance
```

```
!pip install pmdarima
```

##### **3.1.2 Khai báo thư viện**

```
## Library
```

```
import pandas as pd
```

```
import numpy as np
```

```
import yfinance as yf
```

```
import datetime
```

```
from datetime import date
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
import plotly.graph_objects as go
```

```
from statsmodels.tsa.seasonal import seasonal_decompose
```

```
from statsmodels.tsa.stattools import adfuller, kpss
```

```
from statsmodels.tsa.statespace.sarimax import SARIMAX
```

```
from statsmodels.graphics.tsaplots import plot_acf
```

```
from statsmodels.graphics.tsaplots import plot_pacf
```

```
from pmdarima.arima import auto_arima
```

```
from scipy import stats
```

```
import scipy.stats as scs
```

```
import statsmodels.api as sm
from statsmodels.tools.sm_exceptions import InterpolationWarning

import warnings
warnings.filterwarnings("ignore")
```

### 3.1.3 Viết các hàm cần dùng

*# Viết hàm cho kiểm định ADF (ADF\_test):*

Đề kiểm tra phương trình đặc trưng của chuỗi có nghiệm đơn vị hay không chúng ta sử dụng kiểm định ADF.

Giả thuyết null  $H_0$ : được đặt ra đó là phương trình đặc trưng có nghiệm đơn vị. Nếu  $p\text{-value} < 0.1$  thì ta sẽ loại bỏ giả thuyết null, chấp nhận giả thuyết thay thế. Khi đó ta có thể khẳng định rằng chuỗi không có nghiệm đơn vị và có tính chất dừng.

```
def adf_test(x):
    indices = ['Test Statistic', 'p-value', '# of Lags Used', '# of Observations Used']
    adf_test = adfuller(x, autolag='AIC')
    results = pd.Series(adf_test[0:4], index=indices)
    for key, value in adf_test[4].items():
        results[f'Critical Value ({key})'] = value
    return results
```

*# Viết hàm cho kiểm định KPSS (KPSS\_test):*

Kiểm tra KPSS (Kiểm tra Kwiatkowski – Phillips – Schmidt – Shin): kiểm tra một chuỗi thời gian đứng yên xung quanh xu hướng trung bình hoặc tuyến tính, hoặc không đứng yên do một đơn vị gốc.

$H_0$ : quá trình này là ổn định xu hướng

$H_1$ : quá trình có gốc đơn vị

```
def kpss_test(x, h0_type='c'):
    indices = ['Test Statistic', 'p-value', '# of Lags']
    kpss_test = kpss(x, regression=h0_type)
```

```

results = pd.Series(kpss_test[0:3], index=indices)
for key, value in kpss_test[3].items():
    results[f'Critical Value ({key})'] = value
return results

```

## 3.2 Thu thập dữ liệu

### 3.2.1 Tải dữ liệu lên, làm sạch dữ liệu và định dạng cho dữ liệu

*## Tải dữ liệu của CRO*

```
dateparse = lambda dates: pd.datetime.strptime(dates, '%Y-%m-%d')
```

```

df_CRO_RAW = yf.download('CRO-USD',
                           start='2019-6-30',
                           end='2022-1-16',
                           parse_dates=['date'],
                           index_col='date',
                           date_parser=dateparse,
                           progress=False)

```

```
print(f'Downloaded {df_CRO_RAW.shape[0]} rows of data.')
```

```
df_CRO_RAW.to_excel("CRO_RAW.xlsx")
```

```
df_cro = pd.read_excel("CRO_RAW.xlsx")
```

*##Đọc Data*

```
df_cro["Symbol"] = "CRO"
```

```
df_cro
```

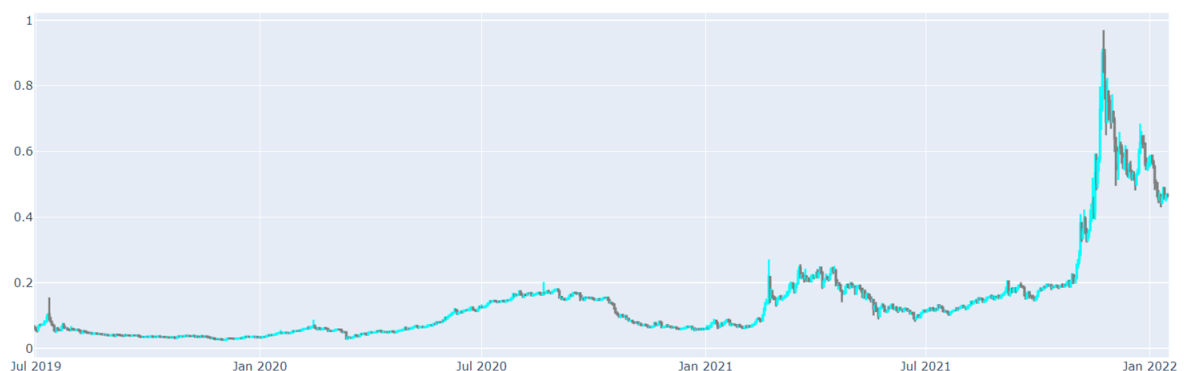


	Date	Open	High	Low	Close	Adj Close	Volume	Symbol
0	2019-06-30	0.063360	0.069947	0.057197	0.061162	0.061162	6550538	CRO
1	2019-07-01	0.061162	0.065025	0.051503	0.056784	0.056784	5771235	CRO
2	2019-07-02	0.056157	0.059115	0.048205	0.055500	0.055500	4888326	CRO
3	2019-07-03	0.055727	0.068474	0.055727	0.067856	0.067856	6178373	CRO
4	2019-07-04	0.067710	0.074430	0.062899	0.071698	0.071698	3615473	CRO
...	...	...	...	...	...	...	...	...
927	2022-01-12	0.458830	0.493290	0.458618	0.486546	0.486546	178944129	CRO
928	2022-01-13	0.486519	0.491958	0.456009	0.456061	0.456061	150766601	CRO
929	2022-01-14	0.456170	0.467545	0.447214	0.456634	0.456634	132753870	CRO
930	2022-01-15	0.456681	0.473361	0.456315	0.467501	0.467501	120922480	CRO
931	2022-01-16	0.467495	0.472171	0.460590	0.463847	0.463847	97833541	CRO

932 rows × 8 columns

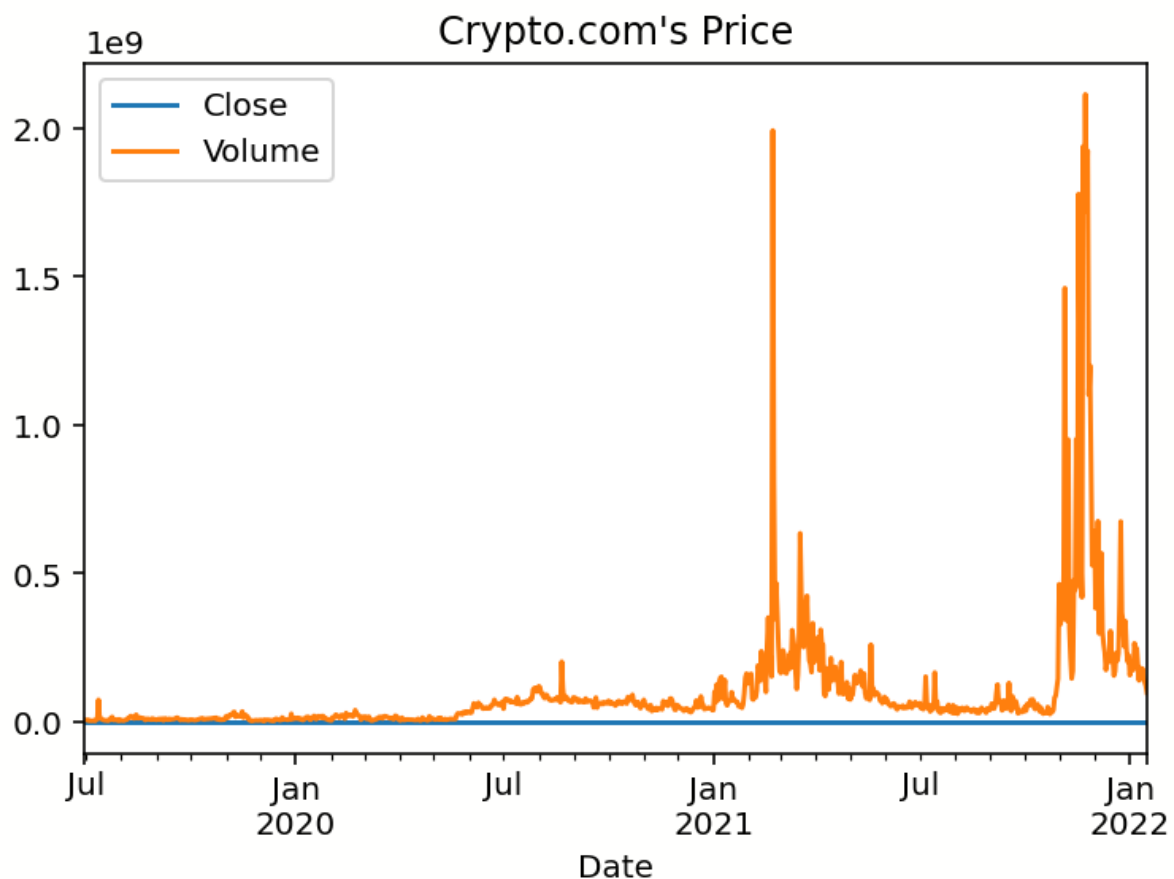
*# Đồ thị hóa các dữ liệu thực bằng candlestick*

```
fig = go.Figure(data=[go.Candlestick(x=df_cro['Date'],
    open=df_cro['Open'],
    high=df_cro['High'],
    low=df_cro['Low'],
    close=df_cro['Close'],
    increasing_line_color= 'cyan', decreasing_line_color= 'gray'))
fig.update_layout(xaxis_rangeslider_visible=False)
fig.show()
```



```
cronos.plot(title="Crypto.com's Price")
```





### 3.2.2 Khảo sát chuỗi thời gian và kiểm tra tính ổn định của dữ liệu

#### a) Khảo sát chuỗi thời gian

*#Tính giá trị trung bình luân phiên theo khoảng thời gian và độ lệch chuẩn của dữ liệu theo tháng và tuần:*

`WINDOW_SIZE = 12 #/ 20 #12 tháng`

```
cronos_m['rolling_mean'] =
cronos_m['Close'].rolling(window=WINDOW_SIZE).mean()
```

```
cronos_m['rolling_std'] = cronos_m['Close'].rolling(window=WINDOW_SIZE).std()
```

```
cronos_m['rolling_mean'].dropna()
```

```
cronos_m['rolling_std'].dropna()
```

*# Trực quan hóa*

```
fig = plt.figure(figsize=(27, 6))
```

```
price = plt.semilogy(cronos_m['Close'], color='blue', label='Original', lw=2, alpha=0.7)
```

```

mean = plt.semilogy(cronos_m['rolling_mean'], color='red', label='Rolling Mean',
lw=2)

std = plt.semilogy(cronos_m['rolling_std'], color='purple', label = 'Rolling Std', lw=2)

plt.legend(loc='best')

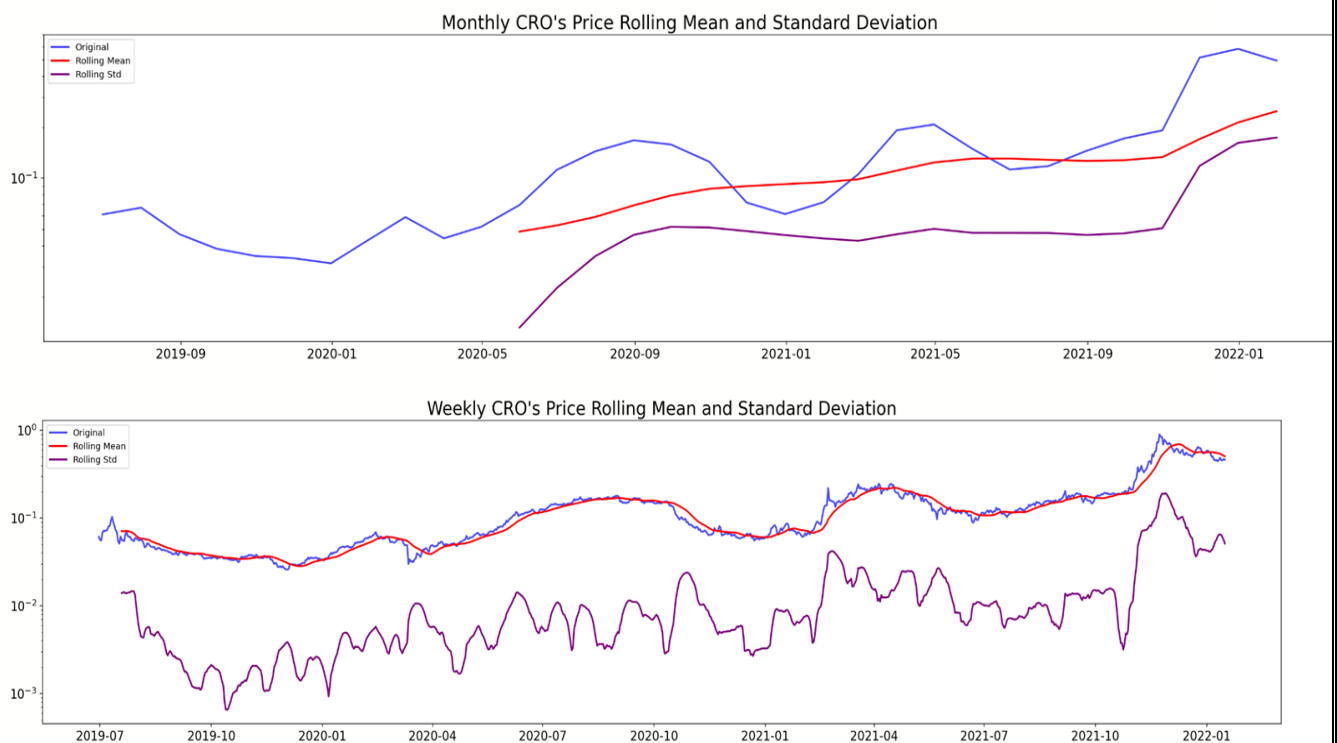
plt.title("Monthly CRO's Price Rolling Mean and Standard Deviation", fontsize=20)

plt.yticks(fontsize=14)

plt.xticks(fontsize=14, rotation=0)

plt.show()

```



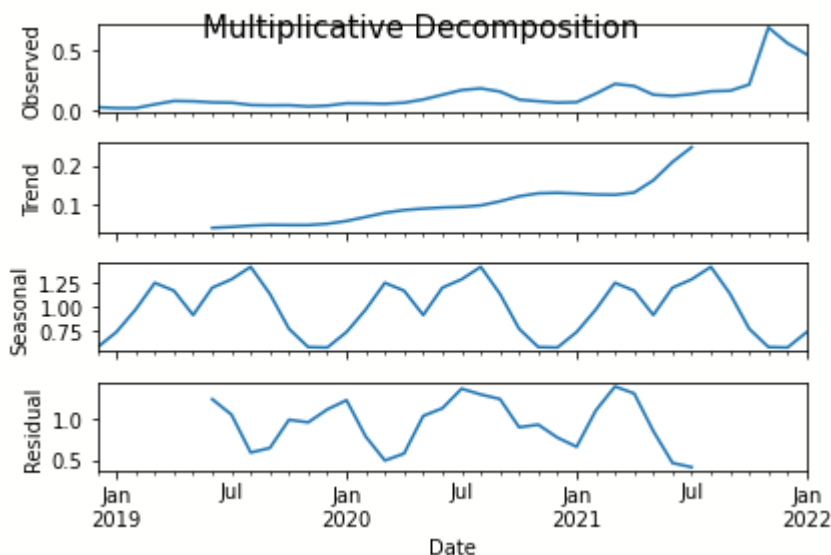
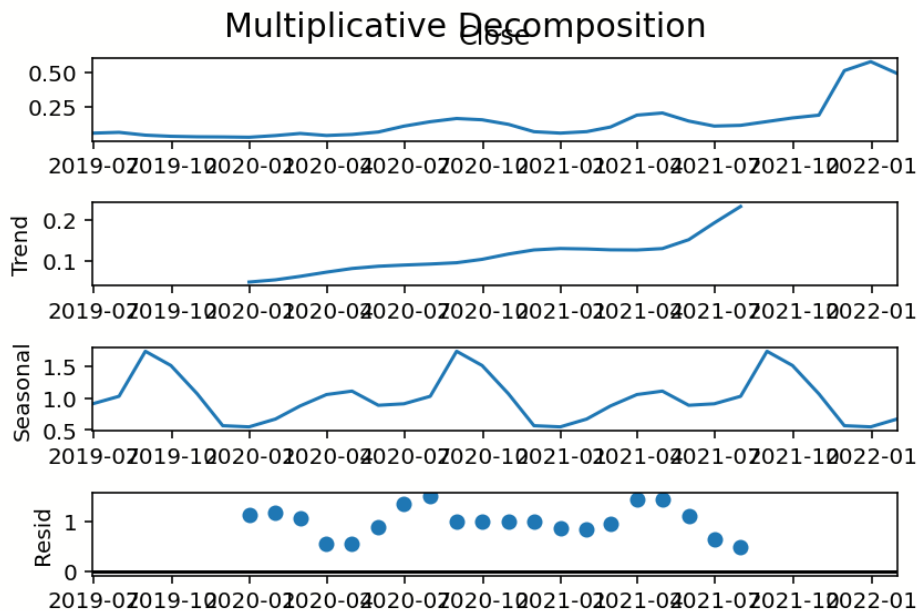
Từ biểu đồ có thể thấy rằng có một mô hình tăng trưởng phi tuyến tính trong giá trị trung bình luân phiên theo khoảng thời gian là 12 tháng và độ lệch chuẩn của dữ liệu tăng theo thời gian, đặc biệt là từ giữa năm 2021. Đó là lý do tại sao em chọn sử dụng mô hình nhân (multiplicative model).

*# Thực hiện phép phân rã mùa vụ*

Việc tìm ra chu kỳ và quy luật mùa vụ sẽ giúp cho mô hình dự báo chuẩn xác hơn. Yếu tố mùa vụ cũng không phải là một trong những yếu tố quá khó nhận biết và có thể dễ dàng phát hiện ra chúng thông qua đồ thị của chuỗi. Như sau

```
decomposition_results =
```

```
seasonal_decompose(cronos_m['Close'],model='multiplicative')
decomposition_results.plot() \
.subtitle('Multiplicative Decomposition',fontsize=15)
```



Biểu đồ phân rã trích xuất ra chuỗi thành phần như: xu hướng (trend), mùa vụ (seasonal), phần dư (residual). Để đánh giá xem phép phân rã có hợp lý hay không, chúng ta có thể xem xét thành phần ngẫu nhiên (residual). Nếu không có mẫu nào rõ ràng (nói cách khác, thành phần ngẫu nhiên thực sự là ngẫu nhiên), thì sự phù hợp có ý nghĩa.

Trong trường hợp dự báo giá Cro, có vẻ như thành phần đã được tách ra khá rõ ràng

như thể hiện trong biểu đồ trên, phép phân rã khá phù hợp và phương sai trong phần còn lại cao hơn một chút trong nửa sau của tập dữ liệu. Cro price chart có tính xu hướng trend, tính mùa vụ seasonal

### **b) Kiểm tra tính dừng**

# KPSS và ADF Test được sử dụng để kiểm tra tính ổn định trong dữ liệu

# Kiểm định ADF

Test Statistic	0.678318
p-value	0.989395
# of Lags Used	9.000000
# of Observations Used	22.000000
Critical Value (1%)	-3.769733
Critical Value (5%)	-3.005426
Critical Value (10%)	-2.642501
dtype: float64	

Vì  $p\text{-value} = 0.989395 > 0.05$  (lớn hơn giá trị tới hạn của mức độ tin cậy đã chọn) nên kết luận rằng chuỗi không đứng yên, chuỗi dữ liệu không có tính chất dừng.

# Kiểm định KPSS

Test Statistic	0.609438
p-value	0.021778
# of Lags	3.000000
Critical Value (10%)	0.347000
Critical Value (5%)	0.463000
Critical Value (2.5%)	0.574000
Critical Value (1%)	0.739000
dtype: float64	

Vì  $p\text{-value} = 0.021778 < 0.1$ , bé hơn giá trị tới hạn của mức độ tin cậy đã chọn) nên kết luận rằng chuỗi không đứng yên, chuỗi dữ liệu không có tính chất dừng.

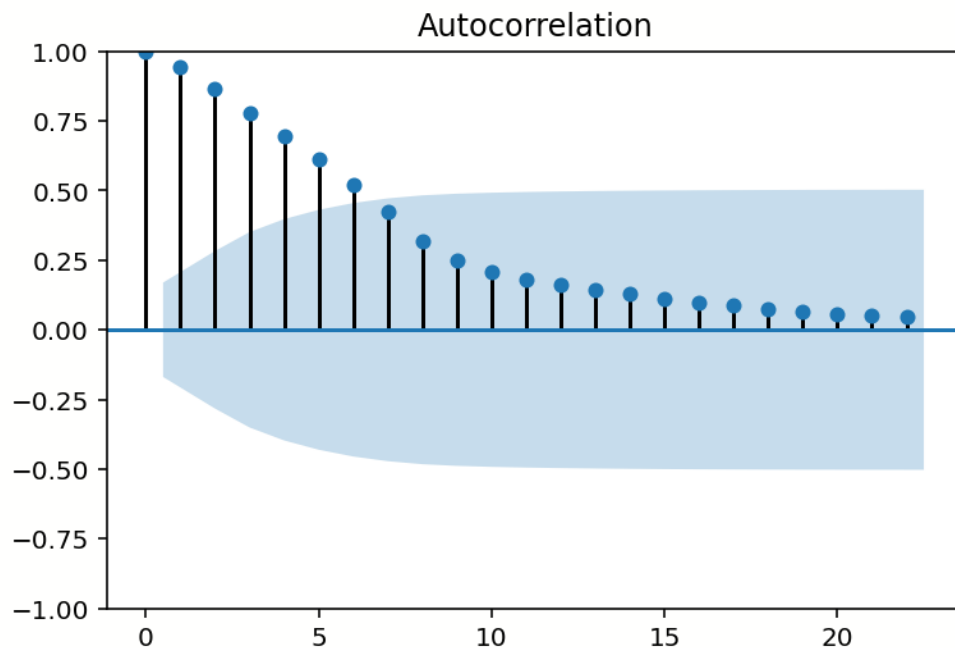
### 3.3 Xây dựng mô hình

#### 3.3.1 Lựa chọn tham số ARIMA(p, d, q)

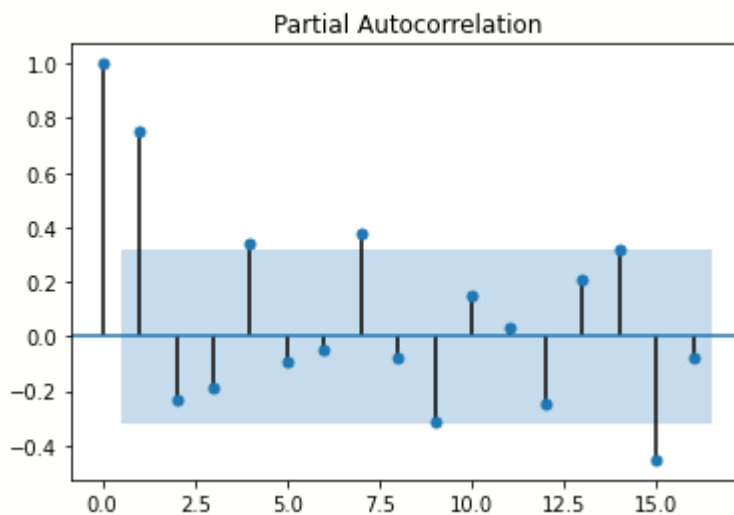
# Kiểm tra ACF (Tự tương quan - AutoCorrelation Function): Mô tả mối tương quan giữa giá trị hiện tại của chuỗi và các giá trị, độ trễ trong quá khứ của nó. Như trình bày ở trên thì chuỗi thời gian có thể được phân tách thành các thành phần như: xu hướng, tính thời vụ, chu kỳ và phần dư. ACF xem xét tất cả các thành phần này để tìm kiếm các mối tương quan do đó nó là một khối tự động tương quan hoàn chỉnh .

# Kiểm tra độ trễ PACF(PACF - Partial AutoCorrelation Function)

Tự động tương quan một phần: Về cơ bản tương quan riêng phần cũng là chỉ số đo lường hệ số tương quan như ACF. Tuy nhiên vẫn có sự khác biệt đó là hệ số tương quan này loại bỏ ảnh hưởng của các chuỗi độ trễ trung gian

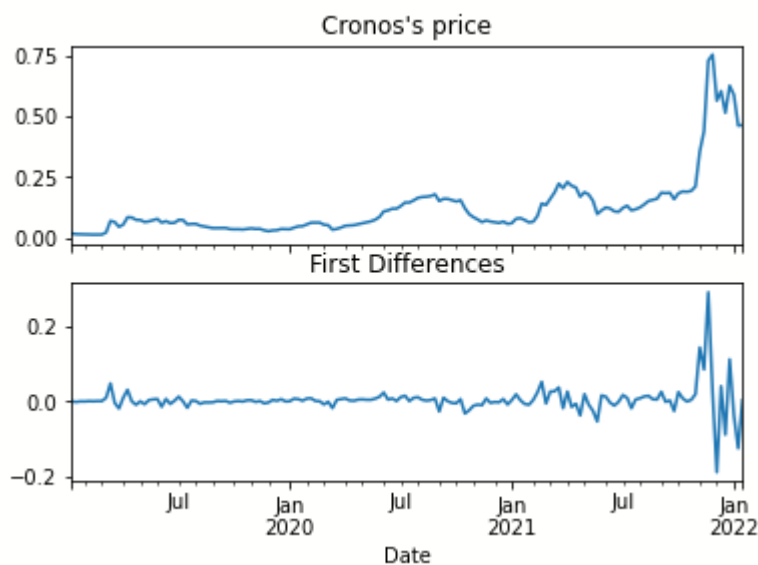


Đối với bài toán này toàn bộ các hệ số tự tương quan với bậc nhỏ hơn hoặc bằng 5 đều có giá trị nằm trong khoảng tin cậy 95% của 0. Do đó, có thể linh hoạt lựa chọn bậc  $q = 5$  là vị trí mà hệ số tự tương quan lớn nhất



Tương tự như ACF, bậc của PACF cũng thường nhỏ hơn 5. Như vậy ta cũng có thể lựa chọn bậc tự do của PACF là một giá trị nào đó từ 1 đến 5. Kết hợp giữa bậc của và giá trị của do chuỗi đã là một chuỗi dừng

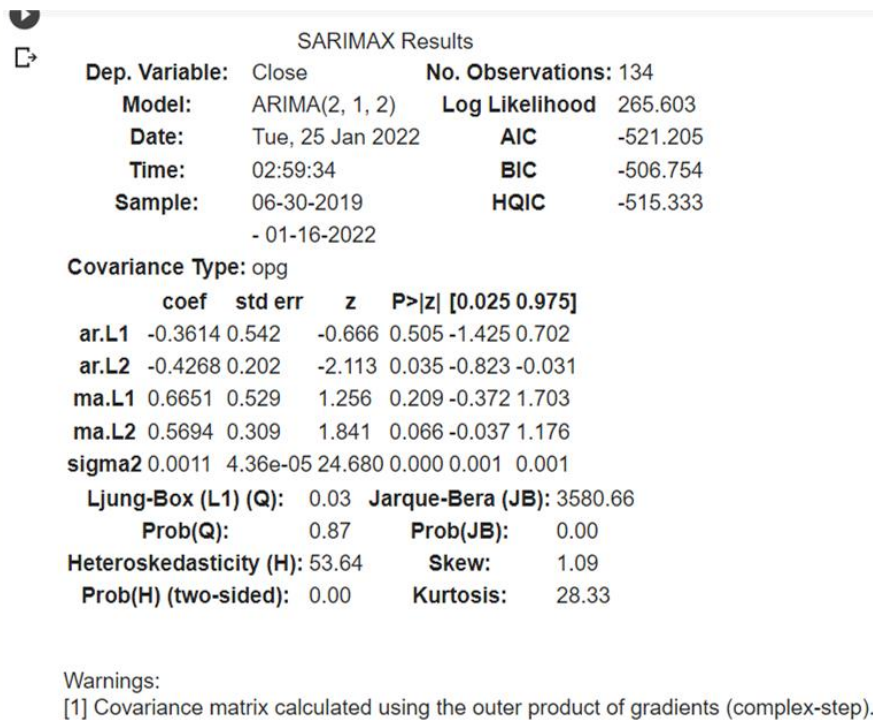
### 3.3.2 Chạy mô hình và phân tích kết quả mô hình ARIMA



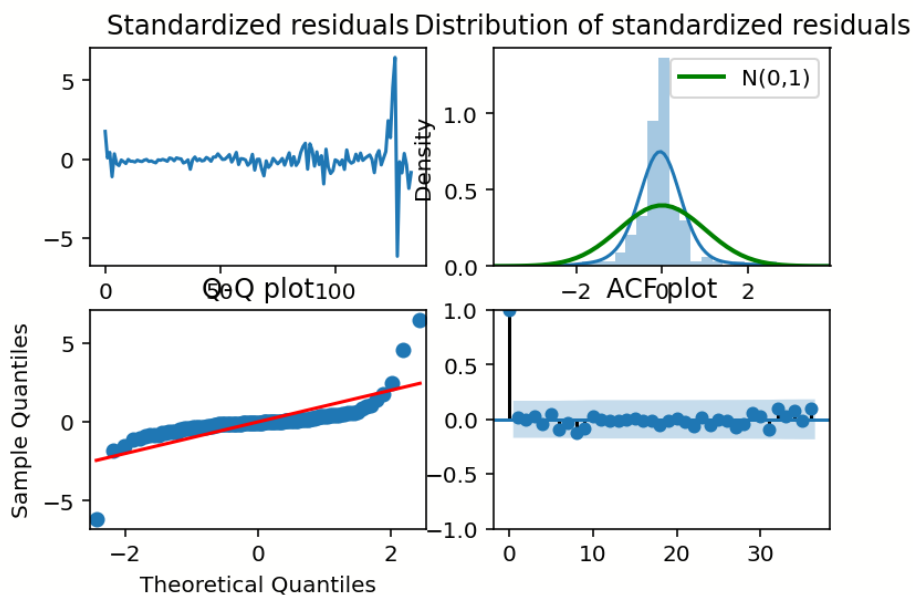
*# Dựa vào kết quả của kiểm định thì ta có thể xác định mô hình ARIMA và fit dữ liệu theo tuần*

*#  $p=q=2$  và  $d = 1$*

```
arima = ARIMA(cronos_w['Close'], order=(2, 1, 2)).fit()
arima.summary()
```



#Kiểm tra phân dư



Biểu đồ cho thấy phân dư được phân phối chuẩn nên có thể nói đây là mô hình khá tốt.

### 3.3.3 Phương pháp Auto ARIMA

Rất khó để xác định đúng các thành phần  $p, d, q$  đúng nên cần có sự trợ giúp của hàm



auto arima để tìm ra mô hình phù hợp nhất. Auto arima trả lại mô hình tốt nhất là:  
Best model: ARIMA(0,1,0)(2,1,1)[12]

```
model_sarima = auto_arima(train, start_p=0, start_q=0,  
                           max_p=5, max_q=5, m=12,  
                           start_P=0, seasonal=True,  
                           d=1, D=1, trace=True,  
                           error_action='ignore',  
                           suppress_warnings=True,  
                           stepwise=True)
```

```
print(model_sarima.aic())
```

```
Performing stepwise search to minimize aic  
ARIMA(0,1,0)(0,1,1)[12] : AIC=inf, Time=1.03 sec  
ARIMA(0,1,0)(0,1,0)[12] : AIC=-544.659, Time=0.11 sec  
ARIMA(1,1,0)(1,1,0)[12] : AIC=-583.905, Time=0.83 sec  
ARIMA(0,1,1)(0,1,1)[12] : AIC=inf, Time=1.46 sec  
ARIMA(1,1,0)(0,1,0)[12] : AIC=-552.525, Time=0.17 sec  
ARIMA(1,1,0)(2,1,0)[12] : AIC=-602.904, Time=1.68 sec  
ARIMA(1,1,0)(2,1,1)[12] : AIC=-617.804, Time=2.82 sec  
ARIMA(1,1,0)(1,1,1)[12] : AIC=inf, Time=1.54 sec  
ARIMA(1,1,0)(2,1,2)[12] : AIC=-614.012, Time=8.71 sec  
ARIMA(1,1,0)(1,1,2)[12] : AIC=inf, Time=8.39 sec  
ARIMA(0,1,0)(2,1,1)[12] : AIC=-618.498, Time=5.75 sec  
ARIMA(0,1,0)(1,1,1)[12] : AIC=inf, Time=5.48 sec  
ARIMA(0,1,0)(2,1,0)[12] : AIC=-602.998, Time=2.75 sec  
ARIMA(0,1,0)(2,1,2)[12] : AIC=-615.189, Time=8.41 sec  
ARIMA(0,1,0)(1,1,0)[12] : AIC=-582.194, Time=0.70 sec  
ARIMA(0,1,0)(1,1,2)[12] : AIC=inf, Time=4.48 sec  
ARIMA(0,1,1)(2,1,1)[12] : AIC=-617.462, Time=8.92 sec  
ARIMA(1,1,1)(2,1,1)[12] : AIC=-618.343, Time=6.47 sec  
ARIMA(0,1,0)(2,1,1)[12] intercept : AIC=-617.295, Time=5.50 sec  
  
Best model: ARIMA(0,1,0)(2,1,1)[12]  
Total fit time: 75.296 seconds  
-618.4977810117836
```



### SARIMAX Results

Dep. Variable: y No. Observations: 134  
Model: SARIMAX(0, 1, 1) Log Likelihood 264.508  
Date: Tue, 25 Jan 2022 AIC -525.016  
Time: 02:59:38 BIC -519.235  
Sample: 0 HQIC -522.667  
- 134

Covariance Type: opg

	coef	std err	z	P> z	[0.025	0.975]
ma.L1	0.3055	0.025	12.453	0.000	0.257	0.354
sigma2	0.0011	3.92e-05	27.951	0.000	0.001	0.001

Ljung-Box (L1) (Q): 0.01 Jarque-Bera (JB): 2586.37  
Prob(Q): 0.92 Prob(JB): 0.00  
Heteroskedasticity (H): 55.86 Skew: 0.94  
Prob(H) (two-sided): 0.00 Kurtosis: 24.52

Warnings:

[1] Covariance matrix calculated using the outer product of gradients (comp

AutoArima chọn các tham số (p,d,q) là (0,1,1) với p-value < 0.1, không bác bỏ H0. Mô hình có ý nghĩa.

### 3.3.4 Hồi quy mô hình SARIMA

#### SARIMAX Results

Dep. Variable: y No. Observations: 123  
Model: SARIMAX(0, 1, 0)x(2, 1, [1], 12) Log Likelihood 313.249  
Date: Sun, 23 Jan 2022 AIC -618.498  
Time: 14:31:18 BIC -607.696  
Sample: 0 HQIC -614.116  
- 123

Covariance Type: opg

	coef	std err	z	P> z	[0.025	0.975]
ar.S.L12	-0.4022	0.119	-3.386	0.001	-0.635	-0.169
ar.S.L24	-0.2347	0.136	-1.726	0.084	-0.501	0.032
ma.S.L12	-0.8037	0.150	-5.366	0.000	-1.097	-0.510

sigma2	0.00021.51e-05	10.4640.000	0.000	0.000
Ljung-Box (L1) (Q):	0.95	Jarque-Bera (JB):	45.29	
Prob(Q):	0.33	Prob(JB):	0.00	
Heteroskedasticity (H):	5.15	Skew:	-0.22	
Prob(H) (two-sided):	0.00	Kurtosis:	6.11	

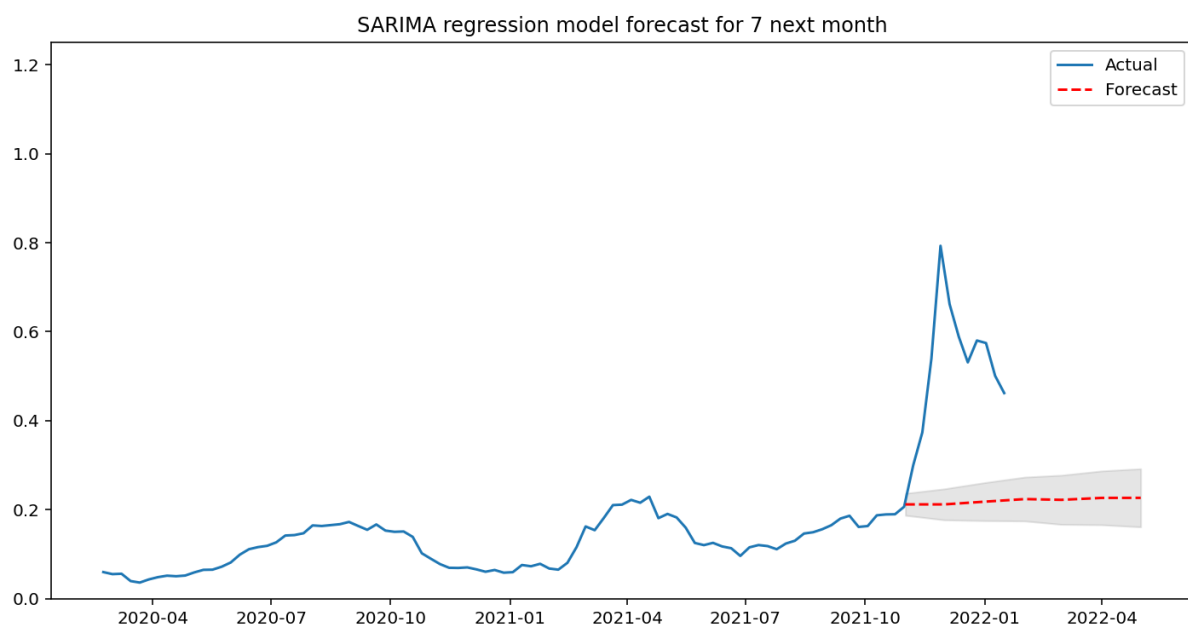
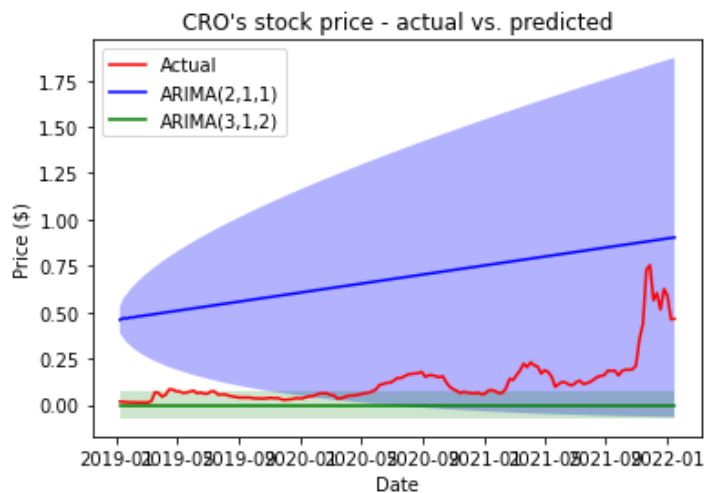
Đây chính là mô hình SARIMA(p=0, d=1, q=0) và (p=2, d=1, q=1,m=12). Mô hình cho kết quả khá tốt khi các hệ số hồi quy đều có ý nghĩa thống kê (toàn bộ cột  $P > |z|$  đều nhỏ hơn 0.1).

### 3.3.5 Kết quả dự báo và thảo luận:

```
n_pred_perious = 7
fitted, confint = model_sarima.predict(n_periods=n_pred_perious, return_conf_int=True)
date = pd.date_range(train.index[-1], periods=n_pred_perious, freq='MS')

fitted_seri = pd.Series(fitted, index=date)
lower = confint[:, 0]
upper = confint[:, 1]

plt.figure(figsize=(12, 6))
plt.plot(cronos_w['Close'][-100:], label='Actual')
plt.plot(fitted_seri, color='red', linestyle='--', label = 'Forecast')
plt.fill_between(date,
                 lower,
                 upper,
                 color='grey', alpha=0.2)
plt.ylim((0, 1.25))
plt.legend()
plt.title('SARIMA regression model forecast for 7 next month')
plt.show()
```



Sau khi đã tìm ra được mô hình ARIMA tốt nhất. Chúng ta sẽ dự báo cho khoảng thời gian tiếp theo. Dự báo cho chuỗi thời gian khá đặc thù và khác biệt so với các lớp mô hình dự báo khác vì giá trị time step liền trước sẽ được sử dụng để dự báo cho time step liền sau. Thông qua hàm predict() ta sẽ dự báo cho 7 tháng tới tương ứng trên tập test kèm theo giá trị ngưỡng tin cậy của nó.

Tuy nhiên, các mô hình lớp ARIMA trong bối cảnh tài chính có điểm yếu là chúng không có khả năng nắm bắt phân nhóm biến động được quan sát thấy trong hầu hết các tài sản tài chính. Như mô hình trên ta thấy mô hình Arima đã không dự báo chính xác giá của CRO, độ lệch khá lớn và nguyên nhân có thể nói đến là do giá của CRO có sự biến động mạnh ở trong khoảng thời gian gần đây.

—

Tham khảo:

phamdinhhkhanh, Mô hình ARIMA trong time series (2019)

Selva Prabhakaran, Augmented Dickey-Fuller Test (ADF Test)

Taniaj, Cryptocurrency Predictions with ARIMA Selva Prabhakaran, ARIMA Model – Complete Guide to Time Series Forecasting in Python (2021) Python | ARIMA Model for Time Series Forecasting (2020)

Eryk Lewinson, Python for Finance Cookbook (2020)

Link:

<https://www.kaggle.com/taniaj/cryptocurrency-predictions-with-arima>

<https://phamdinhhkhanh.github.io/2019/12/12/ARIMAmode.html>

<https://www.youtube.com/watch?v=D9y6dcy0xK8>

<https://www.machinelearningplus.com/time-series/augmented-dickey-fuller-test/>

[https://www.geeksforgeeks.org/python-arima-model-for-time-series-forecasting/?fbclid=IwAR04Jd7pxTwCyIcLLjVWA1RSejIBnh9ylDh9k\\_7P3AogRuSxzvv2xhdeness](https://www.geeksforgeeks.org/python-arima-model-for-time-series-forecasting/?fbclid=IwAR04Jd7pxTwCyIcLLjVWA1RSejIBnh9ylDh9k_7P3AogRuSxzvv2xhdeness)

[https://www.geeksforgeeks.org/python-arima-model-for-time-series-forecasting/?fbclid=IwAR04Jd7pxTwCyIcLLjVWA1RSejIBnh9ylDh9k\\_7P3AogRuSxzvv2xhdeness](https://www.geeksforgeeks.org/python-arima-model-for-time-series-forecasting/?fbclid=IwAR04Jd7pxTwCyIcLLjVWA1RSejIBnh9ylDh9k_7P3AogRuSxzvv2xhdeness)

<https://builtin.com/data-science/time-series-forecasting-python>

<https://www.machinelearningplus.com/time-series/kpss-test-for-stationarity/>

<https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting->

[python/?fbclid=IwAR17tUECQHhGBWKbPY0WmKT3rCXqFzPjI6I5g6\\_wfXaIQkl\\_n8kJoXDMaPVI](https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/?fbclid=IwAR17tUECQHhGBWKbPY0WmKT3rCXqFzPjI6I5g6_wfXaIQkl_n8kJoXDMaPVI)