MÔ TẢ KỊCH BẢN

1. Import thư viện

```
import numpy as np
import pandas as pd
import seaborn as sns #
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)
from sklearn.preprocessing import PowerTransformer
from yellowbrick.cluster import KElbowVisualizer
import lightgbm as lgb
import plotly.graph_objs as go
import plotly.plotly as py
import os
import warnings
```

✓ 0.0s Python

Chọn 3 cột cần làm theo yêu cầu

```
print(2151264683%16+1)
print(2151264683%16+2)
print(2151264683%16+3)
```

✓ 0.0s

```
12
13
14
```

Chuyển đổi cột ngày thành index hiển thị dữ liệu

```
df =pd.read_csv("../data/GiaSMPvaSMPcap2021.csv",encoding="ISO-8859-1",delimiter=";")
df.set_index("Ngay", inplace=True)
df
```

✓ 0.0s Python

Ngay	1	2	3	4	5	6	7	8	9	10	...	39	40
01/01/2021	964.4	964.4	964.4	964.4	964.4	964.4	964.4	964.4	964.4	964.4	...	964.4	964.4
01/02/2021	1019.7	1019.7	1019.7	1019.7	1019.7	1019.7	1019.7	1019.7	1019.7	1019.7	...	1019.7	1019.7
01/03/2021	988.4	988.4	988.4	988.4	988.4	988.4	988.4	988.4	988.4	988.4	...	988.4	988.4
01/04/2021	1002.0	1002.0	1002.0	1002.0	1002.0	1002.0	1002.0	1002.0	1002.1	1002.1	...	1010.8	1010.8
01/05/2021	1061.5	1061.5	1061.5	1061.5	1061.5	1061.5	1061.5	1061.5	1061.5	1061.5	...	1061.5	1061.5
...	...	...	...	...	...	...	...	...	...	...	...	...	...
27/12/2021	1002.0	1002.0	1002.0	1002.0	1002.0	1002.0	1002.0	1002.0	1002.0	1002.0	...	1002.1	1002.1
28/12/2021	1002.0	1002.0	1002.0	1002.0	1002.0	1002.0	1002.0	1002.0	1002.0	1002.0	...	1002.0	1002.0
29/12/2021	1061.5	1061.5	1061.5	1061.5	1061.5	1061.5	1061.5	1061.5	1061.5	1061.5	...	1061.5	1061.5
30/12/2021	1022.6	1022.6	1022.6	1022.6	1022.6	1022.6	1022.6	1022.6	1022.6	1022.6	...	1022.6	1022.6
31/12/2021	1022.6	1022.6	1022.6	1022.6	1022.6	1022.6	1022.6	1022.6	1022.6	1022.6	...	1022.6	1022.6

365 rows × 48 columns

Tạo dataframe gồm 3 cột đã chọn theo yêu cầu

```
feats= ['12','13','14']
```

0.0s

+ Code + Markdown

```
data = df[feats]
data
```

0.0s

Ngay	12	13	14
01/01/2021	964.4	964.4	964.4
01/02/2021	1019.7	1019.7	1019.7
01/03/2021	988.4	988.4	988.4
01/04/2021	1010.8	1010.8	1010.8
01/05/2021	1061.5	1061.5	1061.5
...	...	...	...
27/12/2021	1002.0	1002.0	1002.1
28/12/2021	1002.0	1002.0	1002.0
29/12/2021	1061.5	1061.5	1061.5
30/12/2021	1022.6	1022.6	1022.6
31/12/2021	1022.6	1022.6	1022.6

365 rows × 3 columns

2. Thực hiện EDA đánh giá dữ liệu
   - Trực quan hóa phân phối các cột dữ liệu

Trực quan hóa phân phối các cột dữ liệu

```
# Thống kê mô tả của các cột dữ liệu
print("\nThống kê mô tả của các cột dữ liệu:")
data.describe()
```

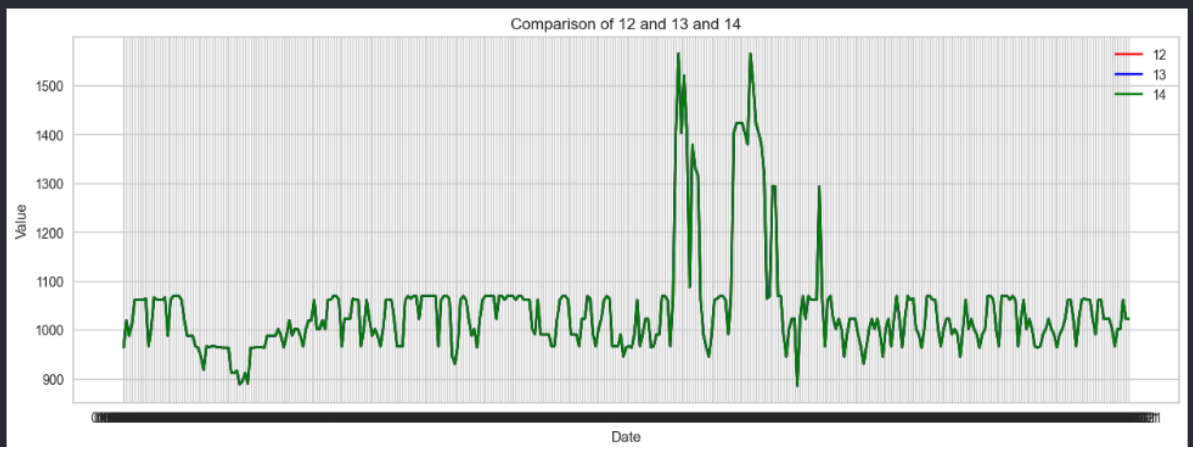
0.0s Python

```
Thống kê mô tả của các cột dữ liệu:
```

	12	13	14
count	365.000000	365.000000	365.000000
mean	1040.311507	1040.312329	1040.314521
std	105.147065	105.146765	105.145787
min	885.700000	885.700000	885.700000
25%	988.400000	988.400000	988.400000
50%	1022.600000	1022.600000	1022.600000
75%	1061.500000	1061.500000	1061.600000
max	1565.500000	1565.500000	1565.500000

```
plt.figure(figsize=(15,5))
plt.plot(data['12'],color = 'red',label = '12')
plt.plot(data['13'],color = 'blue',label = '13')
plt.plot(data['14'],color = 'green',label = '14')
plt.xlabel('Date')
plt.ylabel('Value')
plt.title('Comparison of 12 and 13 and 14')
plt.legend()
plt.show()
```

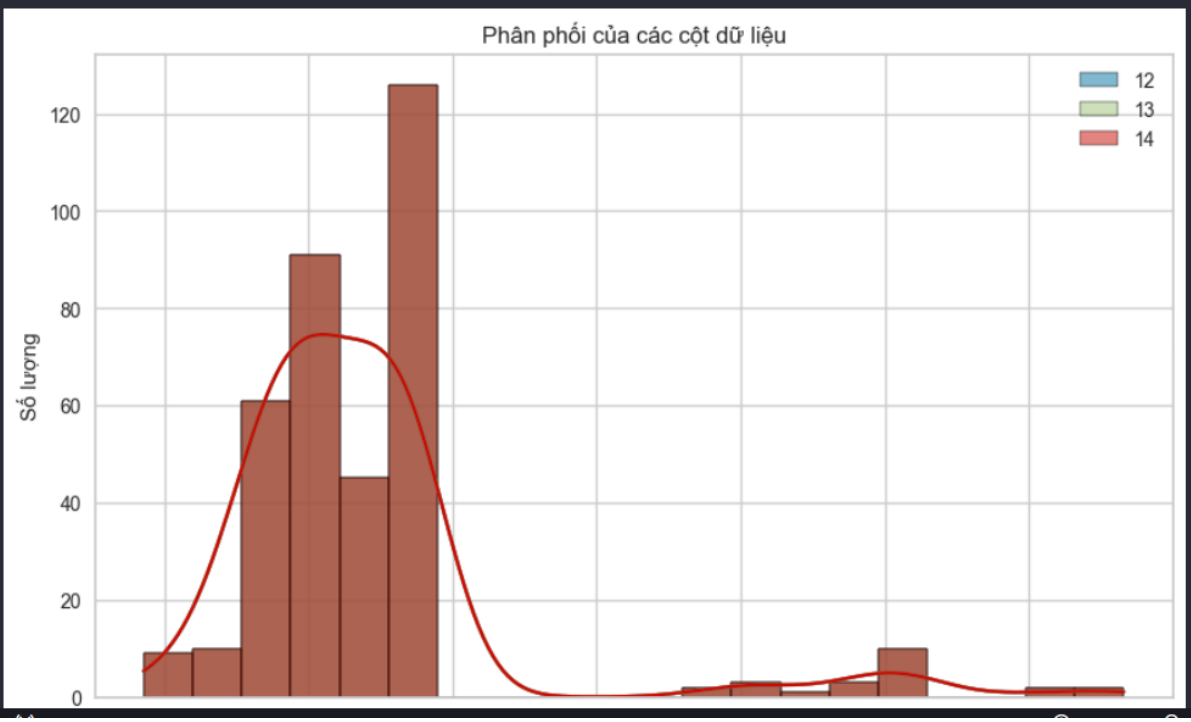
1.2s Python



Nhận xét do các cột dữ liệu quá sát nhau chỉ lệch khoảng 0.01 trên giá trị có độ lớn 1000 nên các biểu đồ gần như là giống nhau và bị đè nên ta chỉ thấy một biểu đồ

```python
# Trực quan hóa phân phối của các cột dữ liệu
plt.figure(figsize=(10, 6))
for column in data.columns:
    sns.histplot(data[column], bins=20, kde=True, label=column)
plt.xlabel("Giá tiền điện")
plt.ylabel("Số lượng")
plt.title("Phân phối của các cột dữ liệu")
plt.legend()
plt.show()
```

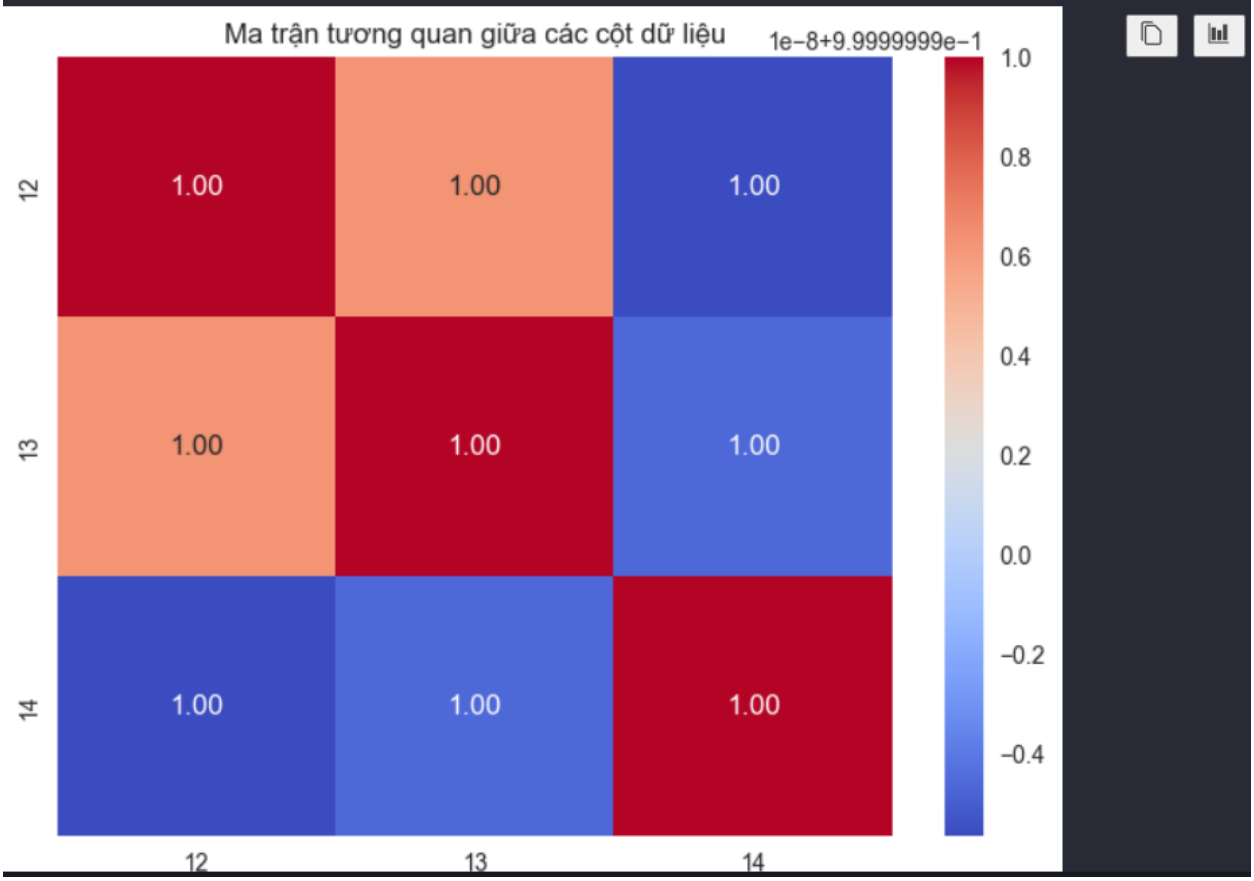
✓ 0.2s Python



Biểu đồ heatmap

```
# Kiểm tra mối quan hệ giữa các cột dữ liệu
plt.figure(figsize=(8, 6))
sns.heatmap(data.corr(), annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Ma trận tương quan giữa các cột dữ liệu")
plt.show()
```

✓ 0.1s



Đánh giá các thuộc tính có độ tương quan khá rời rạc và không thật sự có ảnh hưởng tới nhau

- Kiểm tra dữ liệu thiếu

Kiểm tra và xử lý dữ liệu còn thiếu.

```
# Kiểm tra dữ liệu thiếu
missing_data = data.isnull().sum()
print("Dữ liệu thiếu:")
print(missing_data)
```

[67] ✓ 0.0s

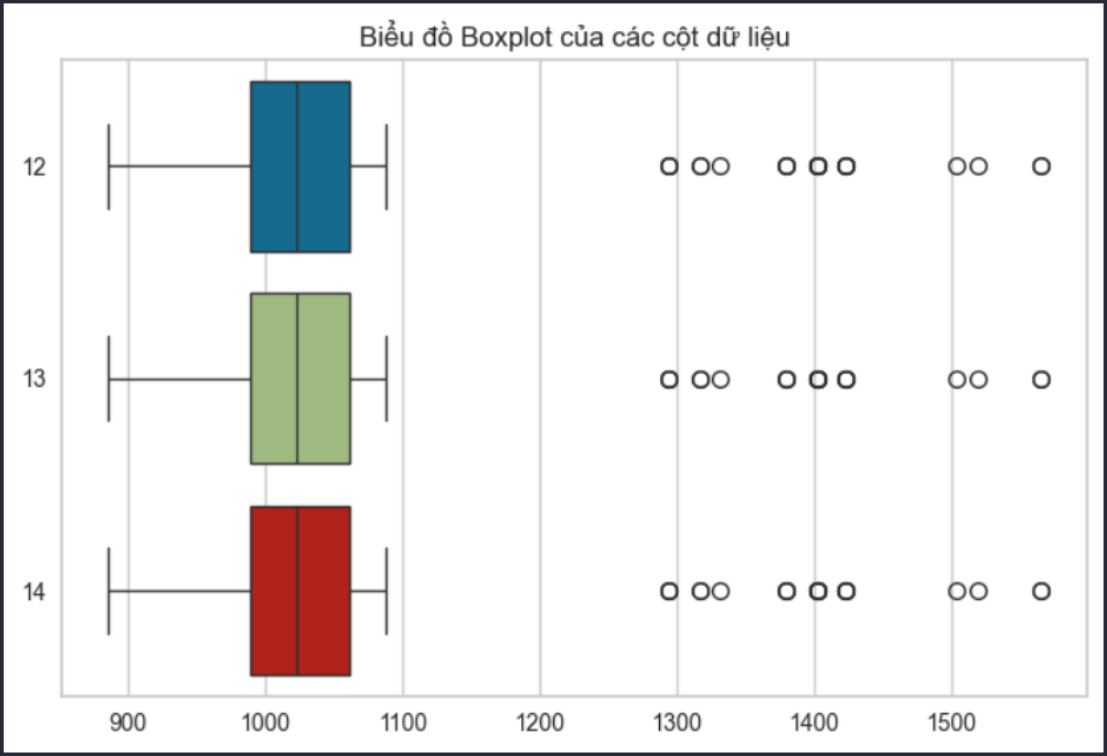
```
Dữ liệu thiếu:
12    0
13    0
14    0
dtype: int64
```

Đánh giá: dữ liệu không thiếu

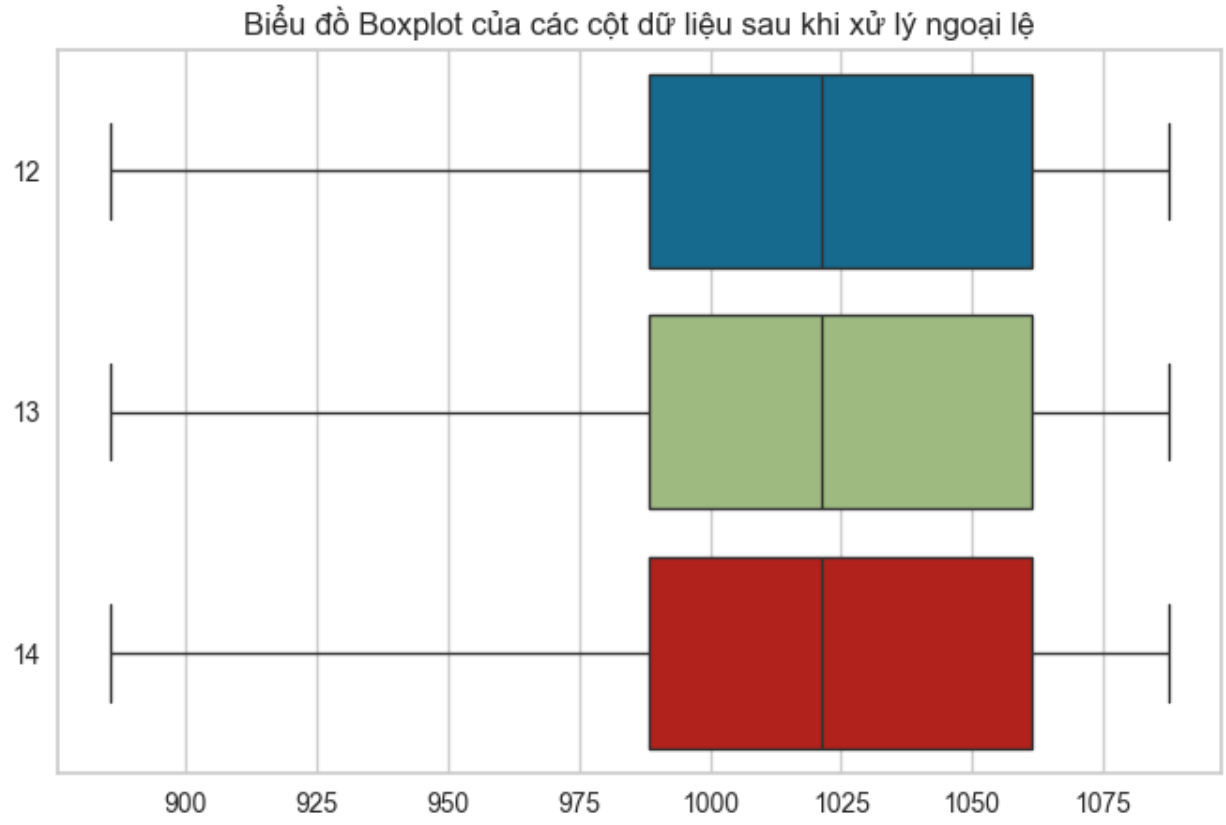
- Xử lý ngoại lệ

```python
# Trực quan hóa boxplot để phát hiện ngoại lệ
plt.figure(figsize=(8, 5))
sns.boxplot(data=data, orient="h")
plt.title("Biểu đồ Boxplot của các cột dữ liệu")
plt.show()
```

✓ 0.1s



Loại bỏ các ngoại lệ



3. Mô hình Kalman

Xây dựng mô hình Kalman cho từng cột dữ liệu điện.

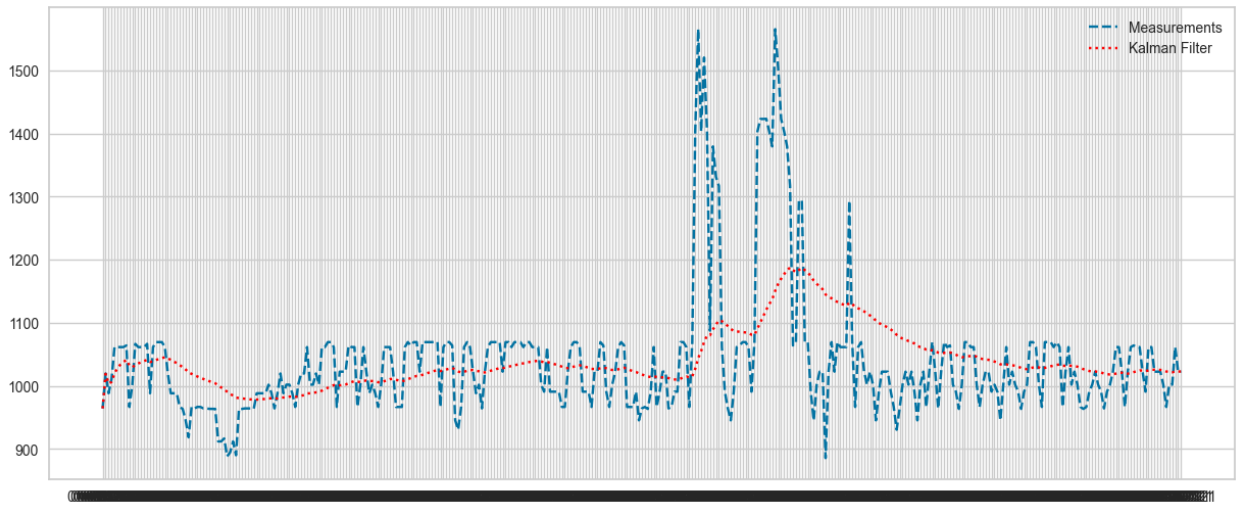
```python
import numpy as np
import matplotlib.pyplot as plt

# Định nghĩa các tham số
Q = 1e-5  # Hiệp phương sai của quá trình (process variance)
R = 0.01  # Hiệp phương sai của đo lường (measurement variance)
n_timesteps = len(data['12'])
xhat = np.zeros(n_timesteps)  # Ước lượng trạng thái ban đầu (a posteri estimate of x)
P = np.zeros(n_timesteps)  # Ước lượng hiệp phương sai của lỗi (a posteri error estimate)
xhatminus = np.zeros(n_timesteps)  # Trạng thái dự đoán (a priori estimate of x)
Pminus = np.zeros(n_timesteps)  # Dự đoán hiệp phương sai của lỗi (a priori error estimate)
K = np.zeros(n_timesteps)  # Kalman gain

# Khởi tạo giá trị ban đầu
xhat[0] = data['12'][0]  # data['12'] là cột dữ liệu cần dự đoán
P[0] = 1.0

# Kalman filter
for k in range(1, n_timesteps):
    # Dự đoán (predict)
    xhatminus[k] = xhat[k-1]
    Pminus[k] = P[k-1] + Q

    # Cập nhật (update)
    K[k] = Pminus[k] / (Pminus[k] + R)
    xhat[k] = xhatminus[k] + K[k] * (data['12'][k] - xhatminus[k])
    P[k] = (1 - K[k]) * Pminus[k]
```

Xây dựng mô hình

```python
# Kalman filter
for k in range(1, n_timesteps):
    # Dự đoán (predict)
    xhatminus[k] = xhat[k-1]
    Pminus[k] = P[k-1] + Q

    # Cập nhật (update)
    K[k] = Pminus[k] / (Pminus[k] + R)
    xhat[k] = xhatminus[k] + K[k] * (data['12'][k] - xhatminus[k])
    P[k] = (1 - K[k]) * Pminus[k]

# Dự báo điểm tiếp theo
xhat_next = xhat[-1]
P_next = P[-1] + Q
K_next = P_next / (P_next + R)
prediction = xhat_next

print(f'Dự báo điểm tiếp theo: {prediction}')

# Vẽ kết quả
plt.figure(figsize=(15, 6))
plt.plot(data['12'], label='Measurements', linestyle='dashed')
plt.plot(xhat, label='Kalman Filter', linestyle='dotted',color='red')

plt.legend()
plt.show()
```

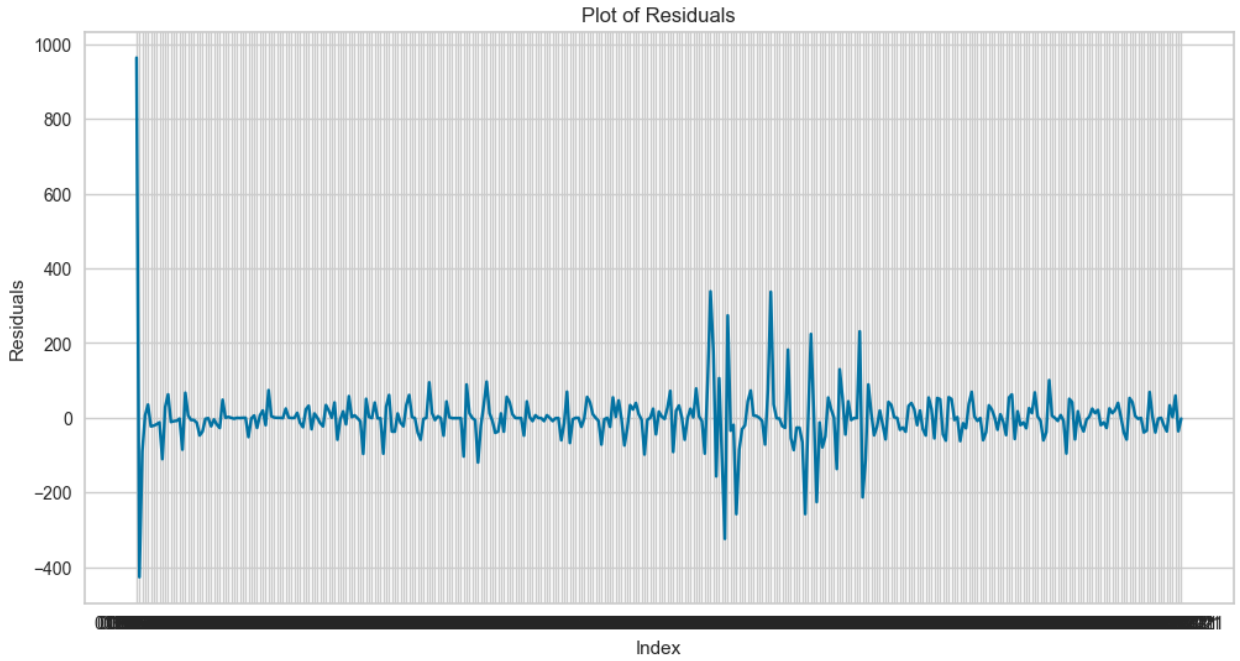
✓ 1.0s Pytho

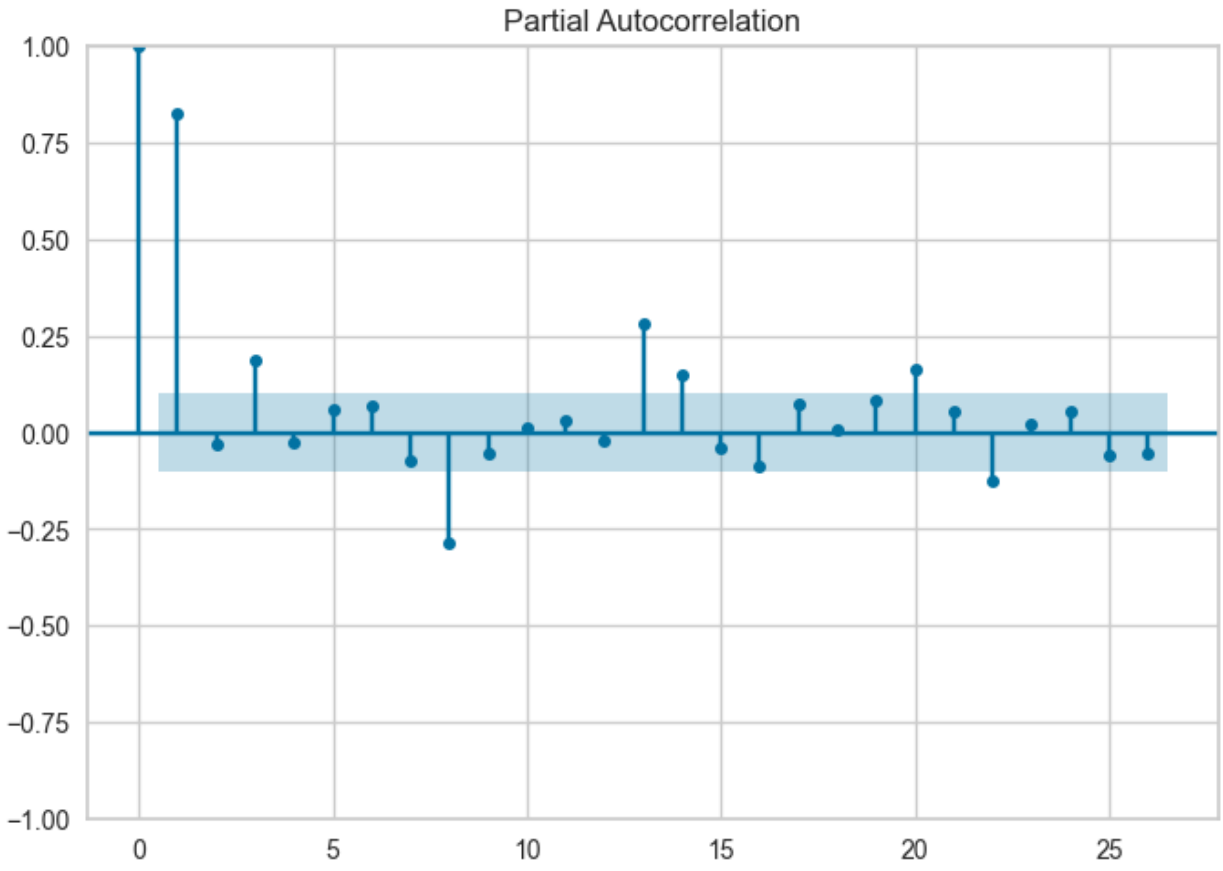
```
C:\Users\Admin\AppData\Local\Temp\ipykernel_18732\2551159785.py:15: FutureWarning:

Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will alwa

C:\Users\Admin\AppData\Local\Temp\ipykernel_18732\2551159785.py:26: FutureWarning:
```

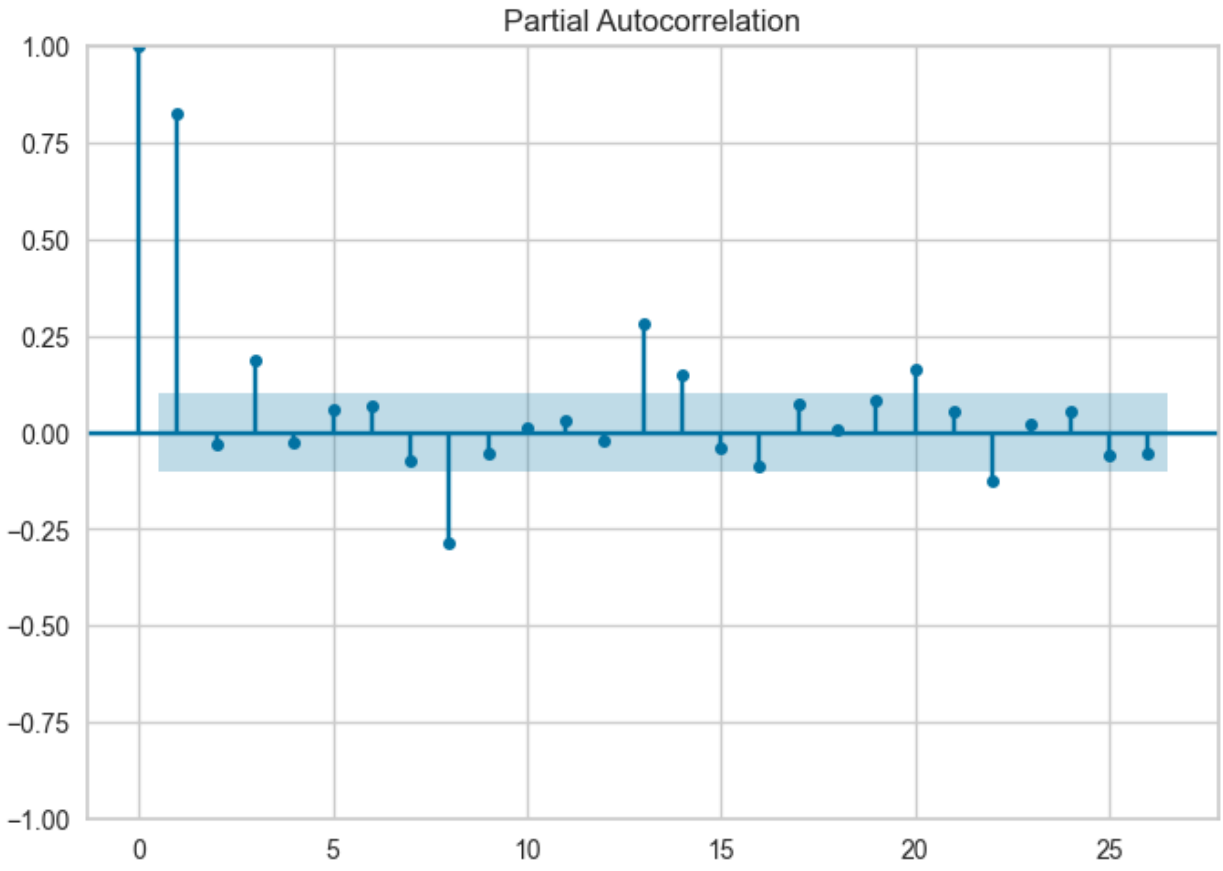


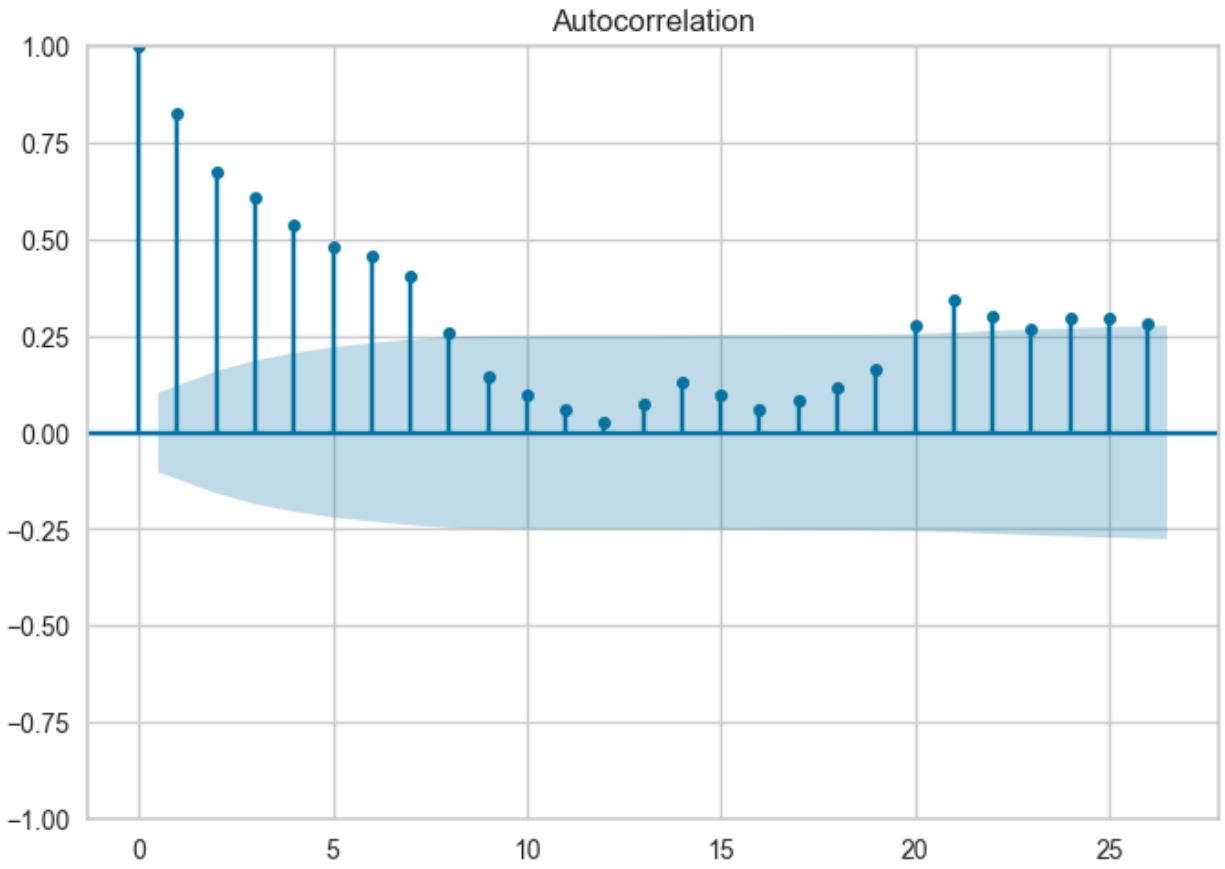
4. Các model khác
   - Mô hình ARIMA

     Trực quan dữ liệu




Partial Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
25


Partial Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
25



Huấn luyện dữ liệu với nhãn là cột 12

```
from statsmodels.tsa.arima.model import ARIMA

# Giả sử bạn chọn p=1, d=1, q=1 sau khi phân tích
model = ARIMA(data['12'], order=(1,1,1))
model_fit = model.fit()
```

✓ 0.1s Pytho

```
C:\Users\Admin\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\loca

An unsupported index was provided and will be ignored when e.g. forecasting.

C:\Users\Admin\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\loca

An unsupported index was provided and will be ignored when e.g. forecasting.

C:\Users\Admin\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\loca

An unsupported index was provided and will be ignored when e.g. forecasting.
```

Dự báo 5 ngày kế tiếp

```python
# Dự báo 5 bước tiếp theo
forecast = model_fit.forecast(steps=5)
print(forecast)
```

✓ 0.0s

```
365    1024.305436
366    1025.688295
367    1026.809591
368    1027.718798
369    1028.456031
Name: predicted_mean, dtype: float64
C:\Users\Admin\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\lo

No supported index is available. Prediction results will be given with an integer index beginning at `

C:\Users\Admin\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\lo

No supported index is available. In the next version, calling this method in a model without a support
```