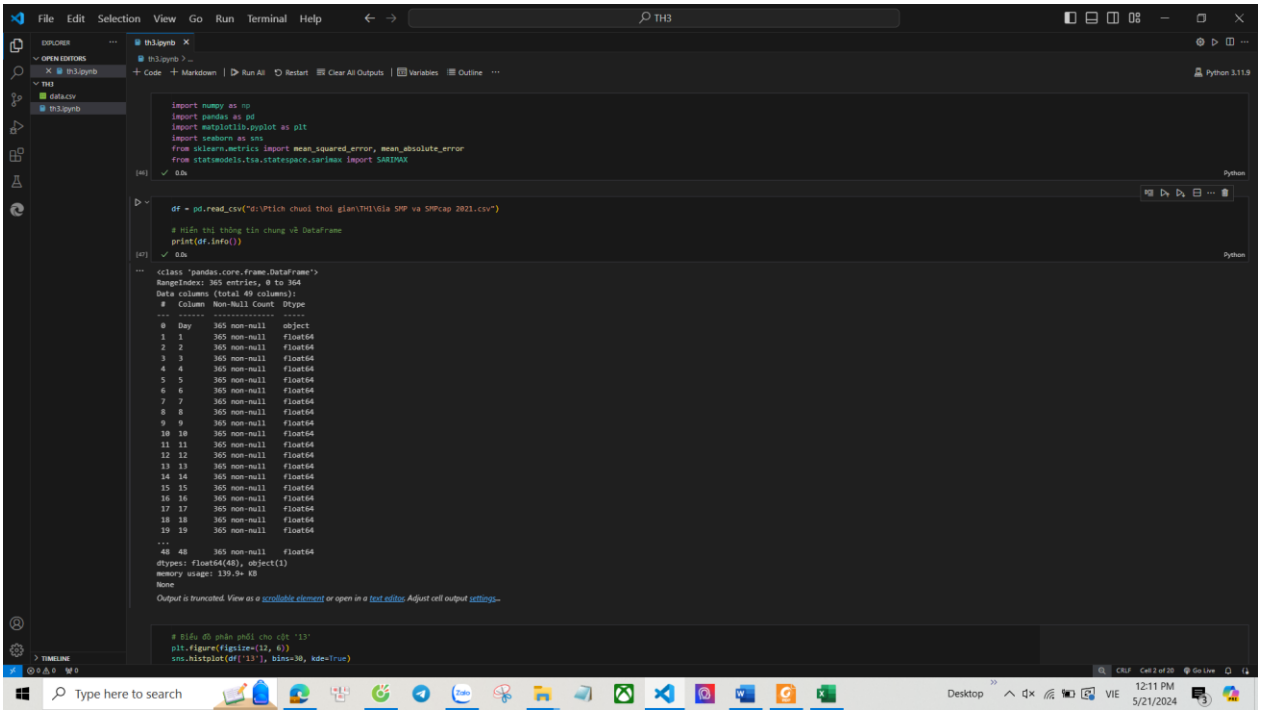
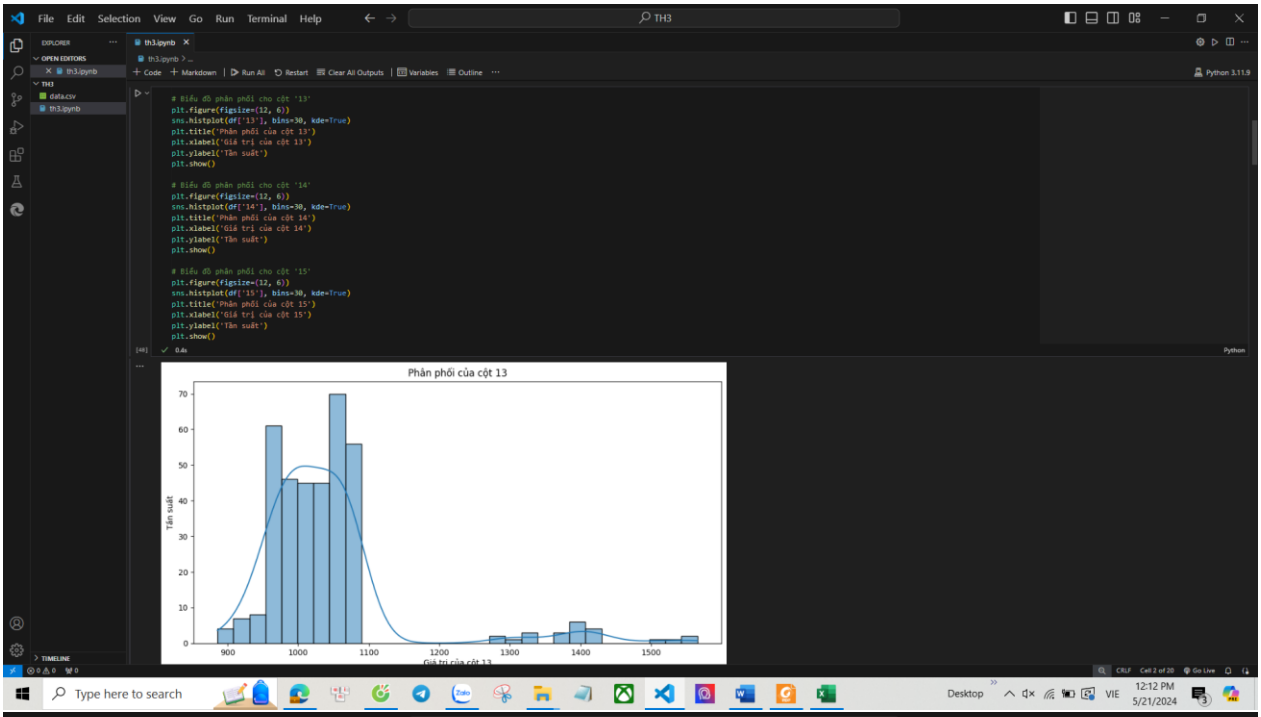
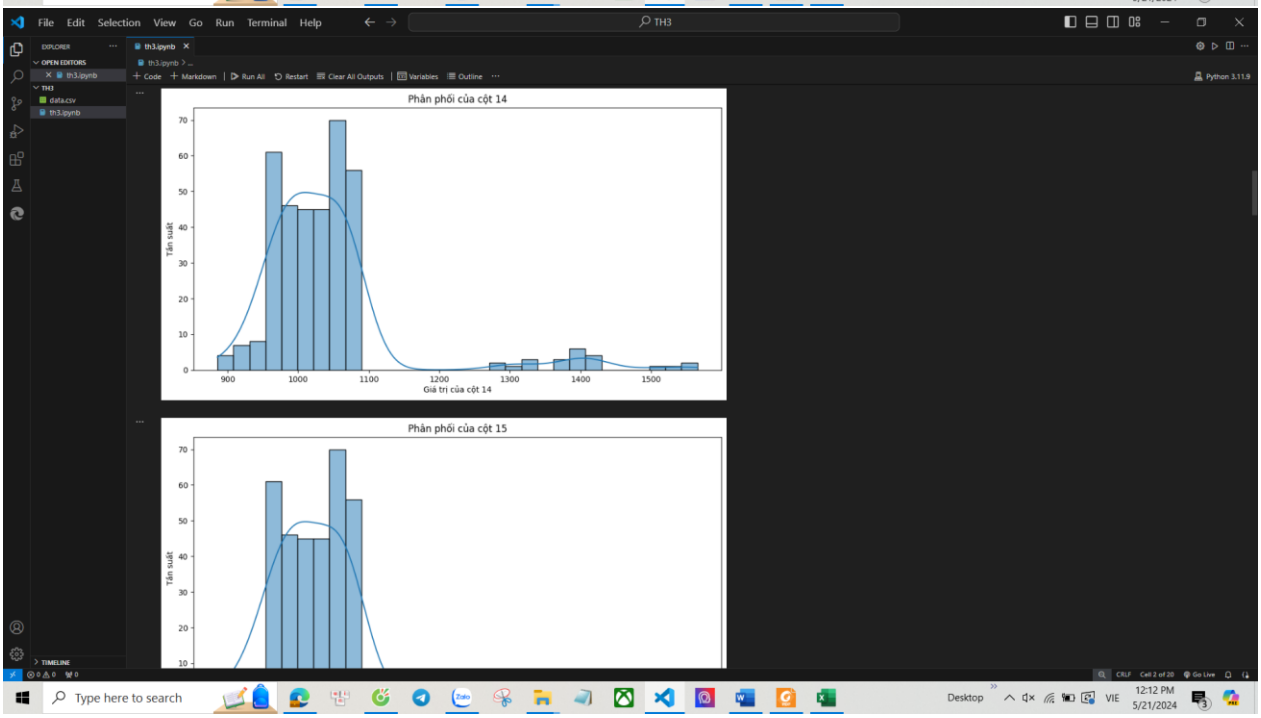
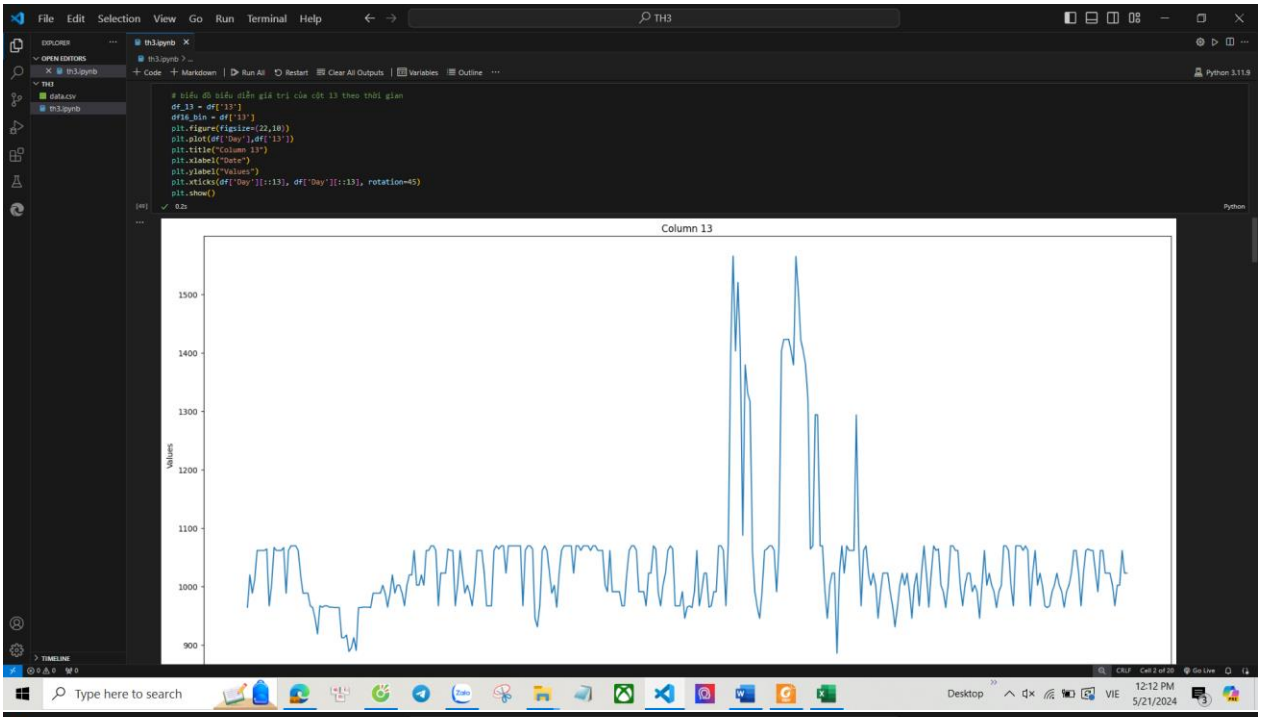
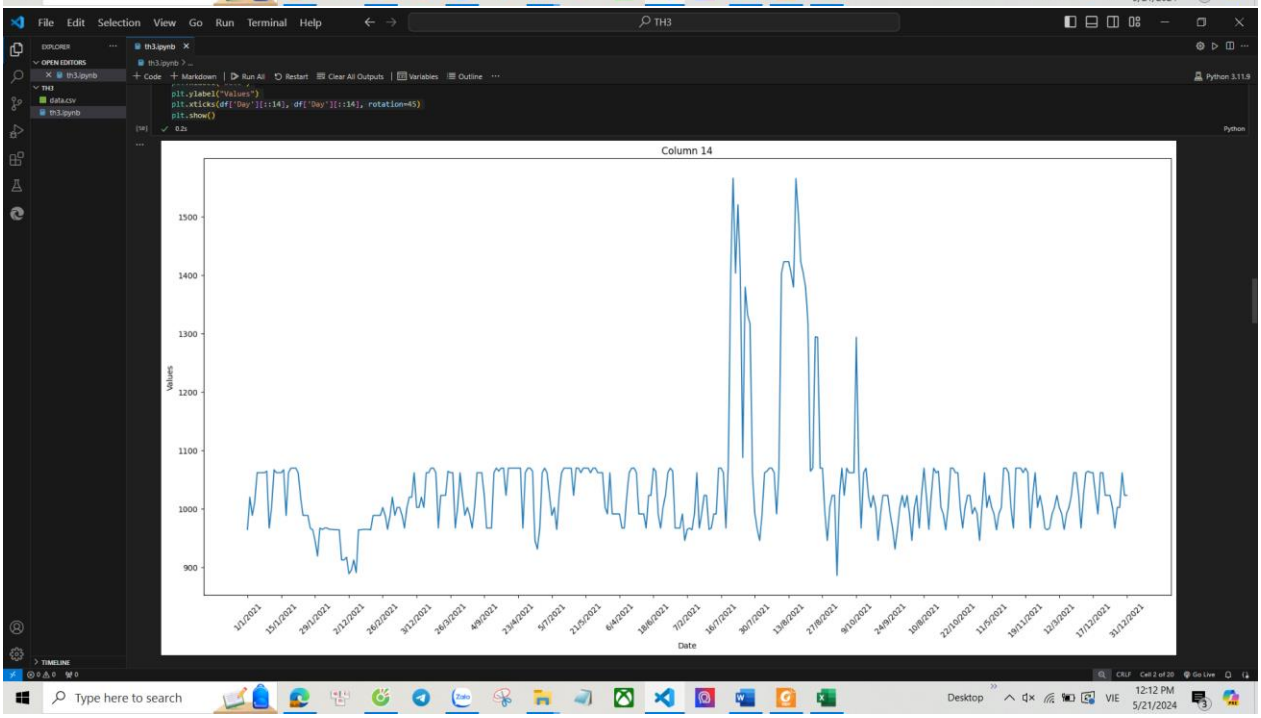
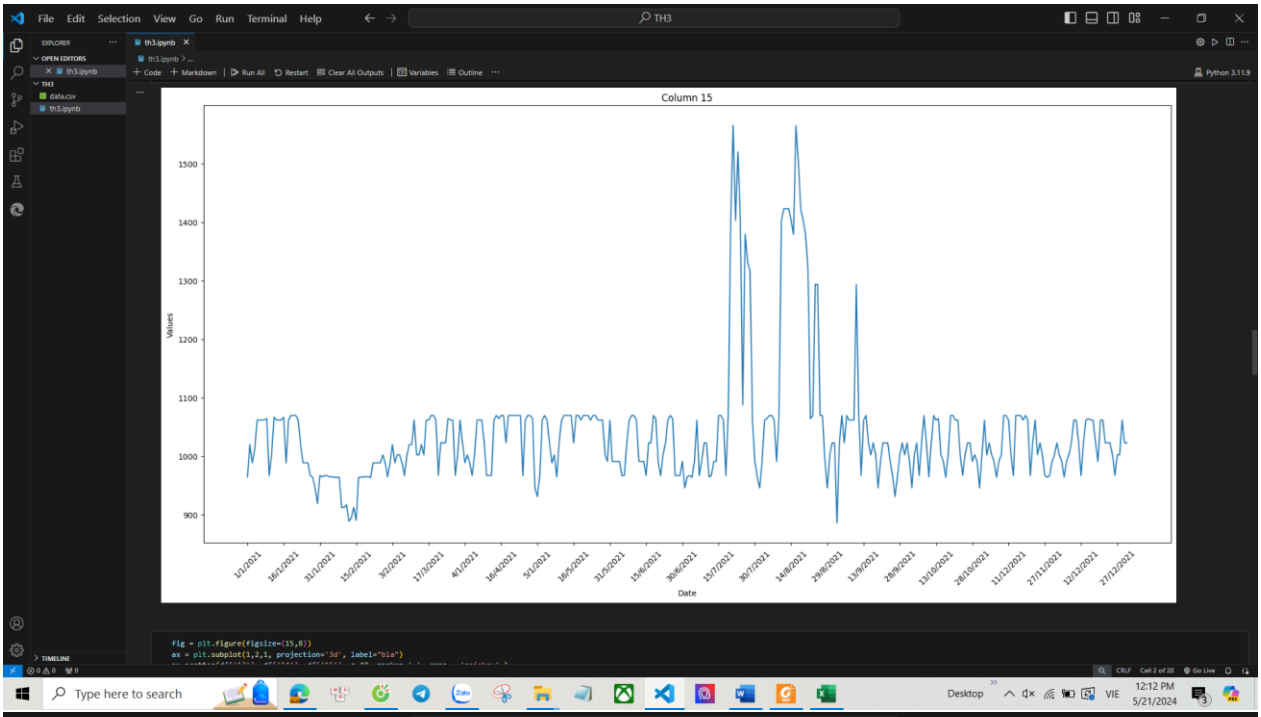

File Edit Selection View Go Run Terminal Help
TH3
th3.ipynb
Code Markdown Run All Restart Clear All Outputs Variables Outline
Python 3.11.9
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import mean_squared_error, mean_absolute_error
from statsmodels.tsa.statespace.sarimax import SARIMAX
0.0s
Python
df = pd.read_csv("d:\Ptich chuoi thoi gian\TH1\Gia SMP va SMPcap 2021.csv")
# Hiển thị thông tin chung về DataFrame
print(df.info())
0.0s
Python
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 49 columns):
Column Non-Null Count Dtype
0 Day 365 non-null object
1 1 365 non-null float64
2 2 365 non-null float64
3 3 365 non-null float64
4 4 365 non-null float64
5 5 365 non-null float64
6 6 365 non-null float64
7 7 365 non-null float64
8 8 365 non-null float64
9 9 365 non-null float64
10 10 365 non-null float64
11 11 365 non-null float64
12 12 365 non-null float64
13 13 365 non-null float64
14 14 365 non-null float64
15 15 365 non-null float64
16 16 365 non-null float64
17 17 365 non-null float64
18 18 365 non-null float64
19 19 365 non-null float64
48 48 365 non-null float64
dtypes: float64(48), object(1)
memory usage: 139.9+ KB
None
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
# Biểu đồ phân phối cho cột '13'
plt.figure(figsize=(12, 6))
sns.histplot(df['13'], bins=30, kde=True)
Type here to search
Desktop
VIE
12:11 PM
5/21/2024

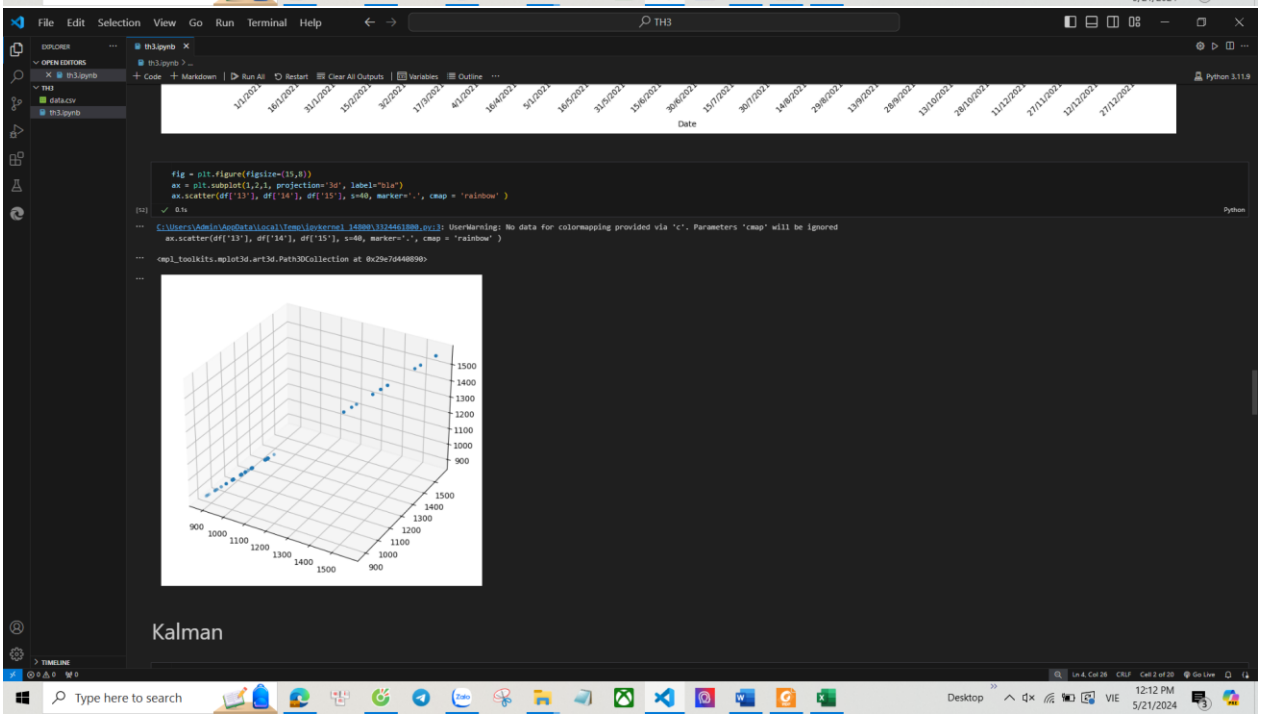

File Edit Selection View Go Run Terminal Help
TH3
th3.ipynb
Code Markdown Run All Restart Clear All Outputs Variables Outline
Python 3.11.9
# Biểu đồ phân phối cho cột '13'
plt.figure(figsize=(12, 6))
sns.histplot(df['13'], bins=30, kde=True)
plt.title('Phân phối của cột 13')
plt.xlabel('Giá trị của cột 13')
plt.ylabel('Tần suất')
plt.show()
# Biểu đồ phân phối cho cột '14'
plt.figure(figsize=(12, 6))
sns.histplot(df['14'], bins=30, kde=True)
plt.title('Phân phối của cột 14')
plt.xlabel('Giá trị của cột 14')
plt.ylabel('Tần suất')
plt.show()
# Biểu đồ phân phối cho cột '15'
plt.figure(figsize=(12, 6))
sns.histplot(df['15'], bins=30, kde=True)
plt.title('Phân phối của cột 15')
plt.xlabel('Giá trị của cột 15')
plt.ylabel('Tần suất')
plt.show()
Phân phối của cột 13
Tần suất
Giá trị của cột 13
Type here to search
Desktop
VIE
12:12 PM
5/21/2024

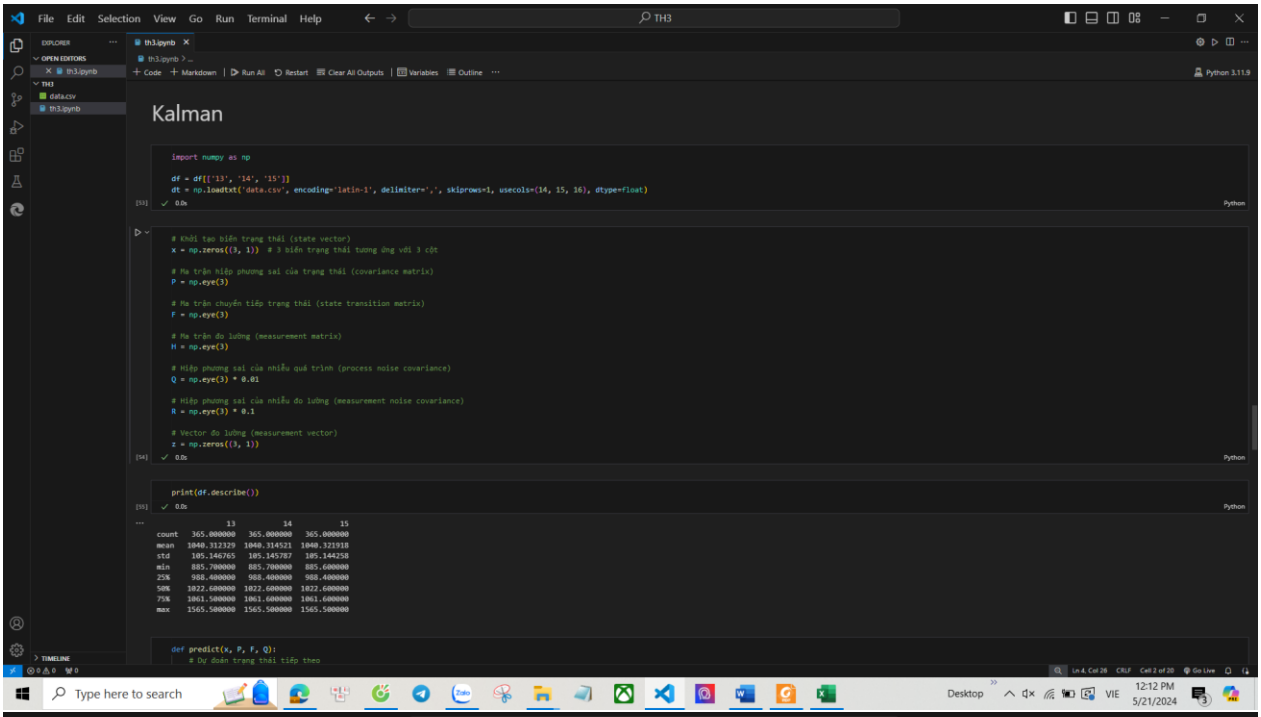

File Edit Selection View Go Run Terminal Help
TH3
th3.ipynb
Code Markdown Run All Restart Clear All Outputs Variables Outline
Python 3.11.9
Phân phối của cột 14
Tần suất
Giá trị của cột 14
Phân phối của cột 15
Tần suất
Type here to search
Desktop
VIE
12:12 PM
5/21/2024

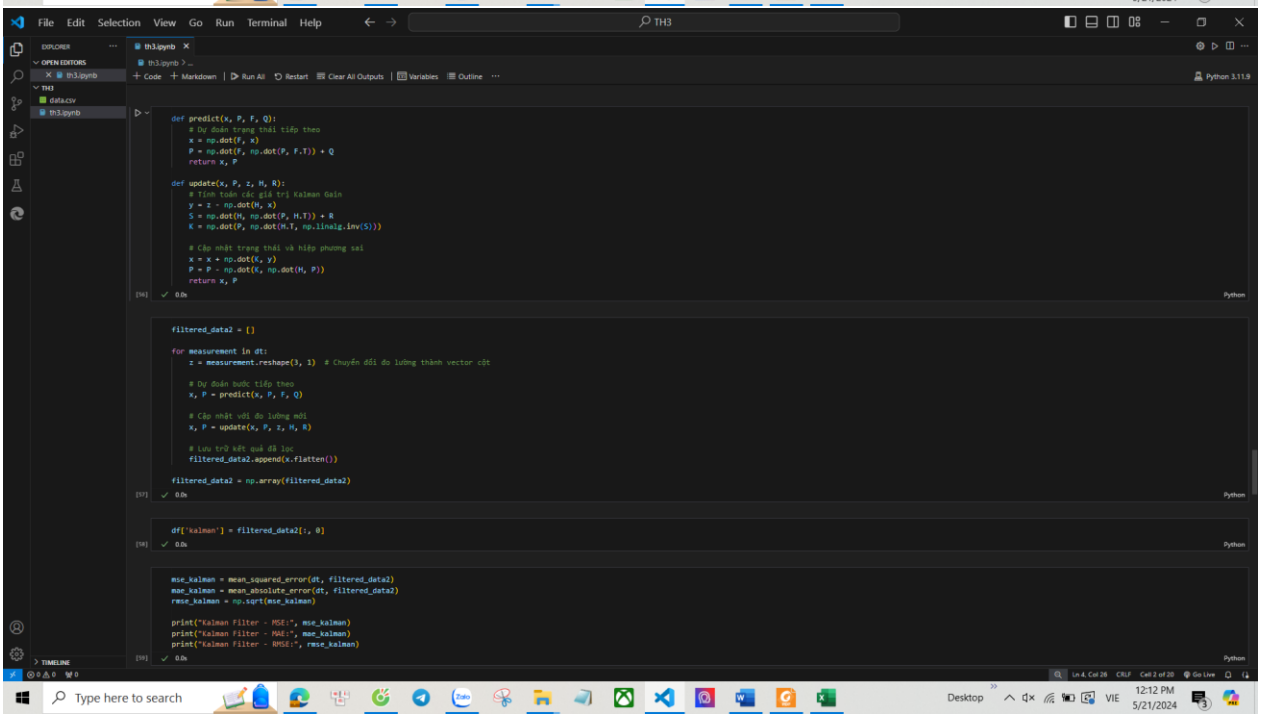

File Edit Selection View Go Run Terminal Help
TH3
th3.ipynb
Code Markdown Run All Restart Clear All Outputs Variables Outline
Python 3.11.9
# biểu đồ biểu diễn giá trị của cột 13 theo thời gian
df_13 = df['13']
df16_bin = df['13']
plt.figure(figsize=(22,10))
plt.plot(df['Day'],df['13'])
plt.title("Column 13")
plt.xlabel("Date")
plt.ylabel("Values")
plt.xticks(df['Day'][::13], df['Day'][::13], rotation=45)
plt.show()
0.2s
Python
Column 13
Values
1500
1400
1300
1200
1100
1000
900
Type here to search
Desktop
VIE
12:12 PM
5/21/2024


File Edit Selection View Go Run Terminal Help
TH3
th3.ipynb
Code Markdown Run All Restart Clear All Outputs Variables Outline
Python 3.11.9
plt.ylabel("Values")
plt.xticks(df['Day'][::14], df['Day'][::14], rotation=45)
plt.show()
0.2s
Python
Column 14
Values
1500
1400
1300
1200
1100
1000
900
1/1/2021
15/1/2021
29/1/2021
12/2/2021
26/2/2021
12/3/2021
26/3/2021
4/4/2021
23/4/2021
5/7/2021
21/5/2021
6/4/2021
18/6/2021
7/2/2021
16/7/2021
30/7/2021
13/8/2021
27/8/2021
9/10/2021
24/9/2021
10/8/2021
22/10/2021
11/5/2021
19/11/2021
12/3/2021
17/12/2021
31/12/2021
Date
Type here to search
Desktop
VIE
12:12 PM
5/21/2024


File Edit Selection View Go Run Terminal Help
TH3
th3.ipynb
Code Markdown Run All Restart Clear All Outputs Variables Outline
Python 3.11.9
Column 15
Values
Date
fig = plt.figure(figsize=(15,8))
ax = plt.subplot(1,2,1, projection='3d', label="bla")
Type here to search
Desktop
VIE
12:12 PM
5/21/2024


File Edit Selection View Go Run Terminal Help
TH3
th3.ipynb
Code Markdown Run All Restart Clear All Outputs Variables Outline
Python 3.11.9
Date
fig = plt.figure(figsize=(15,8))
ax = plt.subplot(1,2,1, projection='3d', label="bla")
ax.scatter(df['13'], df['14'], df['15'], s=40, marker='.', cmap = 'rainbow')
0.1s
Python
C:\Users\Admin\AppData\Local\Temp\ipykernel_14880\3324461808.py:3: UserWarning: No data for colormapping provided via 'c'. Parameters 'cmap' will be ignored
ax.scatter(df['13'], df['14'], df['15'], s=40, marker='.', cmap = 'rainbow')
<mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x29e7d440890>
Kalman
Type here to search
Desktop
VIE
12:12 PM
5/21/2024

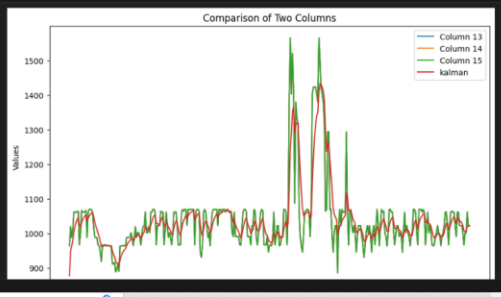

File Edit Selection View Go Run Terminal Help
TH3
th3.ipynb
Code Markdown Run All Restart Clear All Outputs Variables Outline
Python 3.11.9
Kalman
import numpy as np
df = df[['13', '14', '15']]
dt = np.loadtxt('data.csv', encoding='latin-1', delimiter=',', skiprows=1, usecols=(14, 15, 16), dtype=float)
# Khởi tạo biến trạng thái (state vector)
x = np.zeros((3, 1)) # 3 biến trạng thái tương ứng với 3 cột
# Ma trận hiệp phương sai của trạng thái (covariance matrix)
P = np.eye(3)
# Ma trận chuyển tiếp trạng thái (state transition matrix)
F = np.eye(3)
# Ma trận đo lường (measurement matrix)
H = np.eye(3)
# Hiệp phương sai của nhiễu quá trình (process noise covariance)
Q = np.eye(3) * 0.01
# Hiệp phương sai của nhiễu đo lường (measurement noise covariance)
R = np.eye(3) * 0.1
# Vector đo lường (measurement vector)
z = np.zeros((3, 1))
print(df.describe())
13 14 15
count 365.000000 365.000000 365.000000
mean 1040.312329 1040.314521 1040.321918
std 105.146765 105.145787 105.144258
min 885.700000 885.700000 885.600000
25% 988.400000 988.400000 988.400000
50% 1022.600000 1022.600000 1022.600000
75% 1061.500000 1061.600000 1061.600000
max 1565.500000 1565.500000 1565.500000
def predict(x, P, F, Q):
# Dự đoán trạng thái tiếp theo
Type here to search
Desktop
VIE
12:12 PM
5/21/2024


File Edit Selection View Go Run Terminal Help
TH3
th3.ipynb
Code Markdown Run All Restart Clear All Outputs Variables Outline
Python 3.11.9
def predict(x, P, F, Q):
# Dự đoán trạng thái tiếp theo
x = np.dot(F, x)
P = np.dot(F, np.dot(P, F.T)) + Q
return x, P
def update(x, P, z, H, R):
# Tính toán các giá trị Kalman Gain
y = z - np.dot(H, x)
S = np.dot(H, np.dot(P, H.T)) + R
K = np.dot(P, np.dot(H.T, np.linalg.inv(S)))
# Cập nhật trạng thái và hiệp phương sai
x = x + np.dot(K, y)
P = P - np.dot(K, np.dot(H, P))
return x, P
filtered_data2 = []
for measurement in dt:
z = measurement.reshape(3, 1) # Chuyển đổi đo lường thành vector cột
# Dự đoán bước tiếp theo
x, P = predict(x, P, F, Q)
Cập nhật với đo lường mới
x, P = update(x, P, z, H, R)
# Lưu trữ kết quả đã lọc
filtered_data2.append(x.flatten())
filtered_data2 = np.array(filtered_data2)
df['kalman'] = filtered_data2[:, 0]
mse_kalman = mean_squared_error(dt, filtered_data2)
mae_kalman = mean_absolute_error(dt, filtered_data2)
rmse_kalman = np.sqrt(mse_kalman)
print("Kalman Filter - MSE:", mse_kalman)
print("Kalman Filter - MAE:", mae_kalman)
print("Kalman Filter - RMSE:", rmse_kalman)
Type here to search
Desktop
VIE
12:12 PM
5/21/2024

```
mse_kalman = mean_squared_error(dt, filtered_data2)
mae_kalman = mean_absolute_error(dt, filtered_data2)
rmse_kalman = np.sqrt(mse_kalman)

print("Kalman Filter - MSE:", mse_kalman)
print("Kalman Filter - MAE:", mae_kalman)
print("Kalman Filter - RMSE:", rmse_kalman)
```

```
Kalman Filter - MSE: 2795.543314793346
Kalman Filter - MAE: 32.44236277623181
Kalman Filter - RMSE: 52.87289773483143
```

```
# Vẽ biểu đồ so sánh hai cột
plt.figure(figsize=(10, 6))
plt.plot(df['13'], label='Column 13')
plt.plot(df['14'], label='Column 14')
plt.plot(df['15'], label='Column 15')
plt.plot(df['kalman'], label='kalman')
plt.xlabel('Day')
plt.ylabel('Values')
plt.title('Comparison of Two Columns')
plt.legend()
plt.show()
```

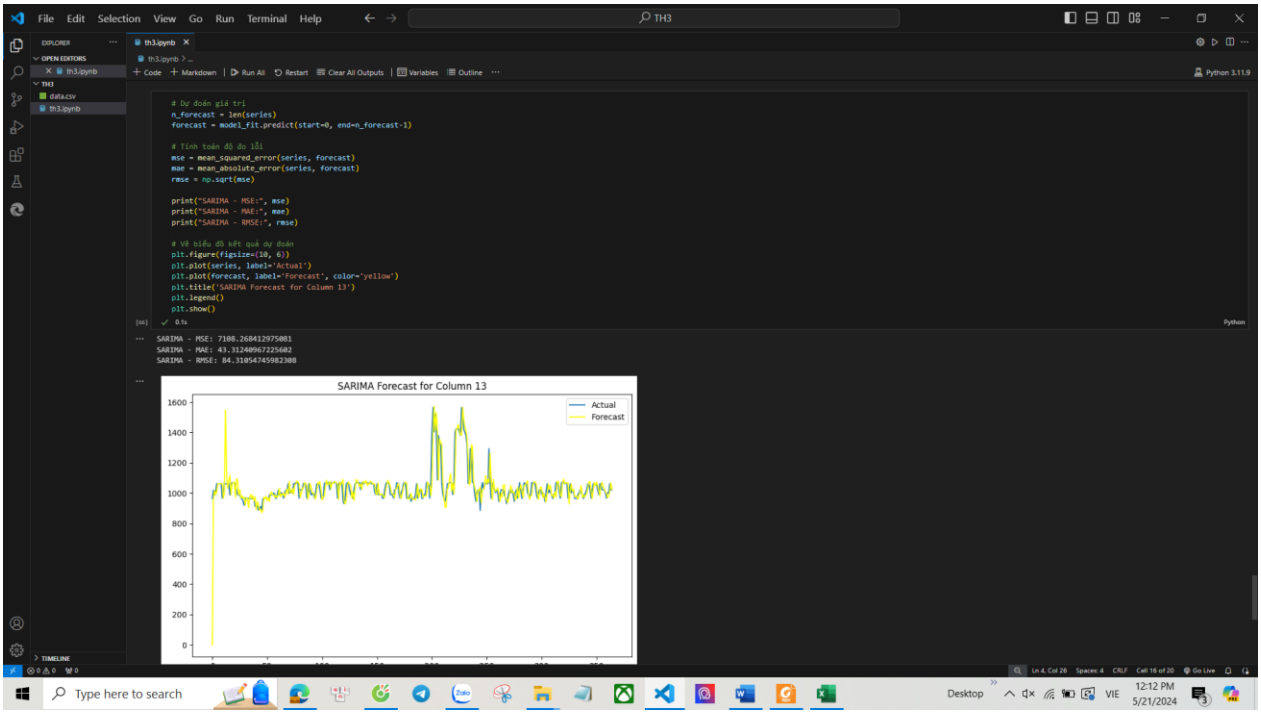


SARIMA

```
df = pd.read_csv('data.csv')
```

```
series = df.iloc[:, 13]

# Điền giá trị thiếu bằng phương pháp nội suy (interpolation)
series = series.interpolate(method='linear')

# Định nghĩa các tham số của mô hình SARIMA
order = (1, 1, 1)       # Tham số (p, d, q) cho phần không mùa
seasonal_order = (1, 1, 1, 12)  # Tham số (P, D, Q, s) cho phần mùa (s = 12 cho dữ liệu theo tháng)

# Xây dựng và huấn luyện mô hình SARIMA
model = SARIMAX(series, order=order, seasonal_order=seasonal_order)
model_fit = model.fit(disp=False)

# In kết quả huấn luyện
print(model_fit.summary())
```

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                                13   No. Observations:                  365
Model:             SARIMAX(1, 1, 1)x(1, 1, 1, 12)   Log Likelihood               -1966.356
Date:                            Tue, 21 May 2024   AIC                           3942.712
Time:                                    12:11:09   BIC                           3962.030
Sample:                                         0   HQIC                          3950.399
                                              - 365
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.9985      9.055      0.110      0.912     -16.748      18.745
ma.L1         -0.9993     17.427     -0.057      0.954     -35.155      33.157
ar.S.L12      -0.2391      0.038     -6.263      0.000      -0.314      -0.164
ma.S.L12      -0.9999     19.280     -0.052      0.959     -38.788      36.788
sigma2      3650.2898   4.79e+04      0.076      0.939   -9.01e+04    9.74e+04
===================================================================================
Ljung-Box (L1) (Q):                   1.29   Jarque-Bera (JB):               981.70
Prob(Q):                              0.26   Prob(JB):                         0.00
Heteroskedasticity (H):               1.62   Skew:                             0.48
Prob(H) (two-sided):                  0.01   Kurtosis:                        11.12
===================================================================================

Warnings:
```


File Edit Selection View Go Run Terminal Help
TH3
th3.ipynb
Code Markdown Run All Restart Clear All Outputs Variables Outline
Python 3.11.9
# Dự đoán giá trị
n_forecast = len(series)
forecast = model_fit.predict(start=0, end=n_forecast-1)
# Tính toán độ đo lỗi
mse = mean_squared_error(series, forecast)
mae = mean_absolute_error(series, forecast)
rmse = np.sqrt(mse)
print("SARIMA - MSE:", mse)
print("SARIMA - MAE:", mae)
print("SARIMA - RMSE:", rmse)
# Vẽ biểu đồ kết quả dự đoán
plt.figure(figsize=(10, 6))
plt.plot(series, label='Actual')
plt.plot(forecast, label='Forecast', color='yellow')
plt.title('SARIMA Forecast for Column 13')
plt.legend()
plt.show()
0.1s
Python
SARIMA - MSE: 7108.268412975081
SARIMA - MAE: 43.31240967225602
SARIMA - RMSE: 84.31054745982308
SARIMA Forecast for Column 13
Actual
Forecast
1600
1400
1200
1000
800
600
400
200
0
Type here to search
Desktop
VIE
12:12 PM
5/21/2024