Đề: 2151260837%3+1=3

Phần 1

Mô hình SARIMA

Tổng quan: SARIMA là một mô hình dự đoán chuỗi thời gian linh hoạt và được sử dụng rộng rãi. Đây là phần mở rộng của mô hình ARIMA không theo mùa, được thiết kế để xử lý dữ liệu theo các mẫu theo mùa. SARIMA nắm bắt cả sự phụ thuộc ngắn hạn và dài hạn trong dữ liệu, khiến nó trở thành một công cụ mạnh mẽ để dự báo. Nó kết hợp các khái niệm về mô hình tự hồi quy, mô hình tích hợp và đường trung bình động với các thành phần theo mùa.

Các thành phần của SARIMA:

1. Thành phần theo mùa: Chữ "S" trong SARIMA thể hiện tính thời vụ, đề cập đến các mẫu lặp lại trong dữ liệu. Đây có thể là hàng ngày, hàng tháng, hàng năm hoặc bất kỳ khoảng thời gian thường xuyên nào khác.
2. Thành phần tự hồi quy (AR): "AR" trong SARIMA biểu thị thành phần tự hồi quy, mô hình hóa mối quan hệ giữa điểm dữ liệu hiện tại và các giá trị trong quá khứ của nó. Nó nắm bắt sự tự tương quan của dữ liệu, nghĩa là mức độ tương quan của dữ liệu với chính nó theo thời gian.
3. Thành phần (I) tích hợp: Chữ "I" trong SARIMA biểu thị sự khác biệt, giúp chuyển đổi dữ liệu không cố định thành dữ liệu cố định. Tính dừng là rất quan trọng đối với mô hình chuỗi thời gian. Thành phần tích hợp đo lường xem cần có bao nhiêu sự khác biệt để đạt được tính ổn định.
4. Thành phần Đường trung bình động (MA): "MA" trong SARIMA đại diện cho thành phần đường trung bình động, mô hình hóa sự phụ thuộc giữa điểm dữ liệu hiện tại và các lỗi dự đoán trong quá khứ. Nó giúp làm mịn chuỗi và loại bỏ các dao động ngẫu nhiên.

Các trường hợp sử dụng SARIMA

Các mô hình SARIMA tìm thấy ứng dụng trong nhiều lĩnh vực khác nhau, bao gồm:

- Kinh tế : Dự đoán các chỉ số kinh tế như lạm phát và GDP.
- Bán lẻ : Dự báo doanh thu và nhu cầu đối với các sản phẩm theo mùa.
- Năng lượng : Dự đoán mức tiêu thụ và nhu cầu năng lượng.

- Chăm sóc sức khỏe : Lập mô hình tiếp nhận bệnh nhân và bùng phát dịch bệnh.
- Tài chính : Dự đoán giá cổ phiếu và xu hướng thị trường.

Mô hình ARIMAX

Tổng quan: Mô hình ARIMAX là phiên bản nâng cao của mô hình ARIMA Mô hình ARIMAX mở rộng khung ARIMA bằng cách tích hợp các biến ngoại sinh, là các yếu tố bên ngoài có thể ảnh hưởng đến chuỗi thời gian đang được nghiên cứu. Việc tích hợp này cho phép mô hình tận dụng thông tin bổ sung có thể nâng cao đáng kể độ chính xác của dự báo.

Các thành phần của mô hình ARIMAX:

ARIMA Model Core : Thành phần ARIMA của mô hình ARIMAX bao gồm 3 phần chính:

- AR (AutoRegressive) : Phần này của mô hình sử dụng sự phụ thuộc giữa một quan sát và một số quan sát bị trễ. Nó giúp hiểu được ảnh hưởng của các giá trị trong quá khứ đến giá trị hiện tại.
- I (Tích hợp) : Điều này liên quan đến việc lấy sai phân chuỗi thời gian để đạt được tính dừng, nghĩa là đảm bảo rằng giá trị trung bình và phương sai không đổi theo thời gian. Tính ổn định là rất quan trọng để dự báo đáng tin cậy.
- MA (Trung bình trượt) : Phần này sử dụng sự phụ thuộc giữa một quan sát và sai số dư từ mô hình trung bình trượt được áp dụng cho các quan sát bị trễ.

Biến ngoại sinh (X) : Đây là những yếu tố hoặc yếu tố dự báo bên ngoài không thuộc chuỗi thời gian nhưng có thể có tác động đáng kể đến chuỗi thời gian. Bằng cách kết hợp các biến này, mô hình ARIMAX có thể cung cấp phân tích toàn diện hơn và hiệu suất dự báo tốt hơn.

Các trường hợp sử dụng ARIMAX

Dự báo tiêu thụ năng lượng: Dự đoán mức tiêu thụ năng lượng (điện, xăng dầu, khí đốt)

Dự báo doanh số bán hàng: Dự báo doanh số bán hàng của các sản phẩm hoặc dịch vụ

Dự báo trong lĩnh vực y tế: Lập mô hình dự báo số lượng bệnh nhân tiếp nhận tại các cơ sở y tế

Dự báo trong lĩnh vực sản xuất và kinh doanh: Dự đoán sản lượng sản xuất hoặc doanh số bán hàng của các công ty

Dự báo trong lĩnh vực kinh tế học: Dự đoán GDP hoặc các chỉ số kinh tế quan trọng

Phần 2

Sử dụng 2 mô hình SARIMA và ARIMAX để dự đoán bài toán

1. Tổng quan về data: dữ liệu gồm 6 cột bao gồm cột date và các cột Truong_1, Truong_2, Truong_3, Truong_4, Truong_5.
2. Tiền xử lý dữ liệu
   - Chuyển định dạng cột ‘date’ về datetime để phục vụ cho bài toán timeseries:

     data['date'] = pd.to_datetime(data['date'], format='%d.%m.%Y')

   - Kiểm tra giá trị thiếu:

     Ta sủ dụng :data.isnull().sum() để kiểm tra cho ra kết quả không có giá trị thiếu nên ko cần tiền xử lý giá trị thiếu

   - Chuẩn hóa số liệu về dang [0,1] giúp cho mô hình dự đoán chính xác hơn
   - Chuẩn hóa các cột số liệu

     scaler = StandardScaler()

     data[['truong_1', 'truong_2', 'truong_3', 'truong_4', 'truong_5']] = scaler.fit_transform(data[['truong_1', 'truong_2', 'truong_3', 'truong_4', 'truong_5']])

   - Kiểm tra trùng lặp ngày :

     Ta sử dụng code sau:

     duplicated_dates = data['date'].duplicated().sum()

     print(f"Number of duplicated dates: {duplicated_dates}")

Kết quả ra :"Number of duplicated dates: 549824"

Vậy nên ta sử dụng các giá trị trung bình cho từng ngày bị trùng lặp và hợp các ngày bị trùng làm 1

- Tính trung bình các giá trị cho mỗi ngày trùng lặp

  data_aggregated = data.groupby('date').mean().reset_index()

  Và cuối cùng chúng ta biết được chuỗi thời gian quan trọng nhất là thời gian phải liên lục. Nhìn bằng mắt thường ta thấy thời gian đang bị rời rạc nên cần bù lại các ngày và các trường bị thiếu. Các trường bị thiếu sẽ được sinh ra bằng các số liệu gần với ngày đó nhất.

- Chuẩn hóa lại các cột số liệu sau khi điền giá trị bị thiếu

  data_filled[['truong_1', 'truong_2', 'truong_3', 'truong_4', 'truong_5']] = scaler.fit_transform(data_filled[['truong_1', 'truong_2', 'truong_3', 'truong_4', 'truong_5']])

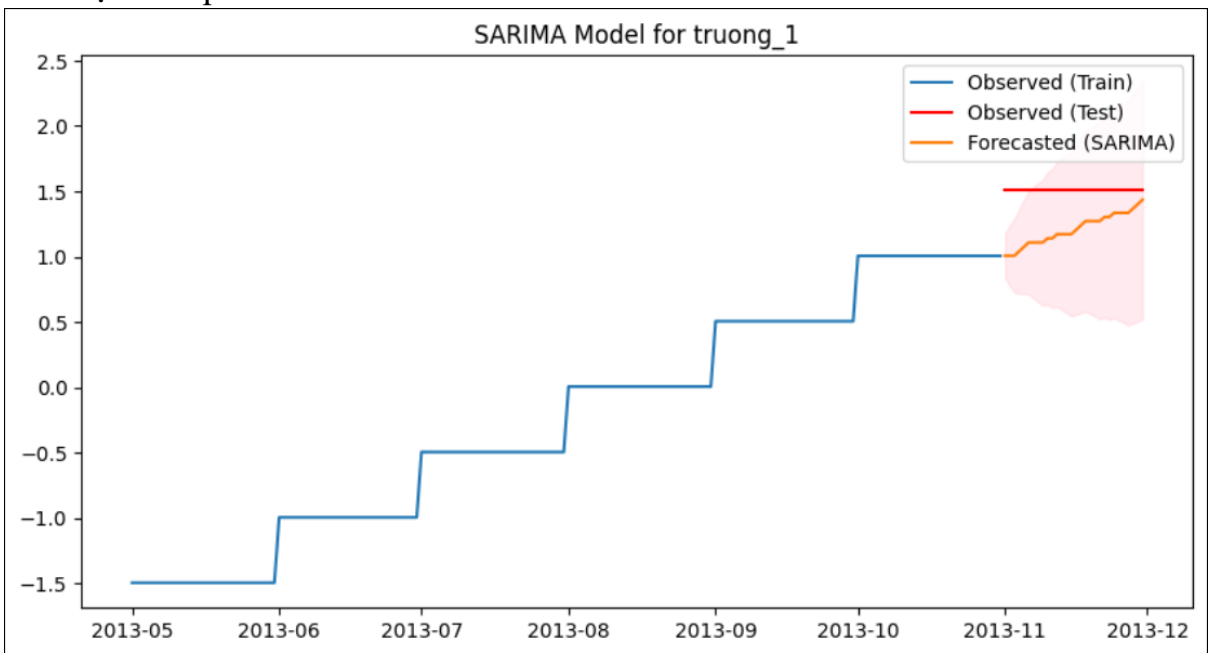
3. Model
   - Tách dữ liệu thành phần train và test

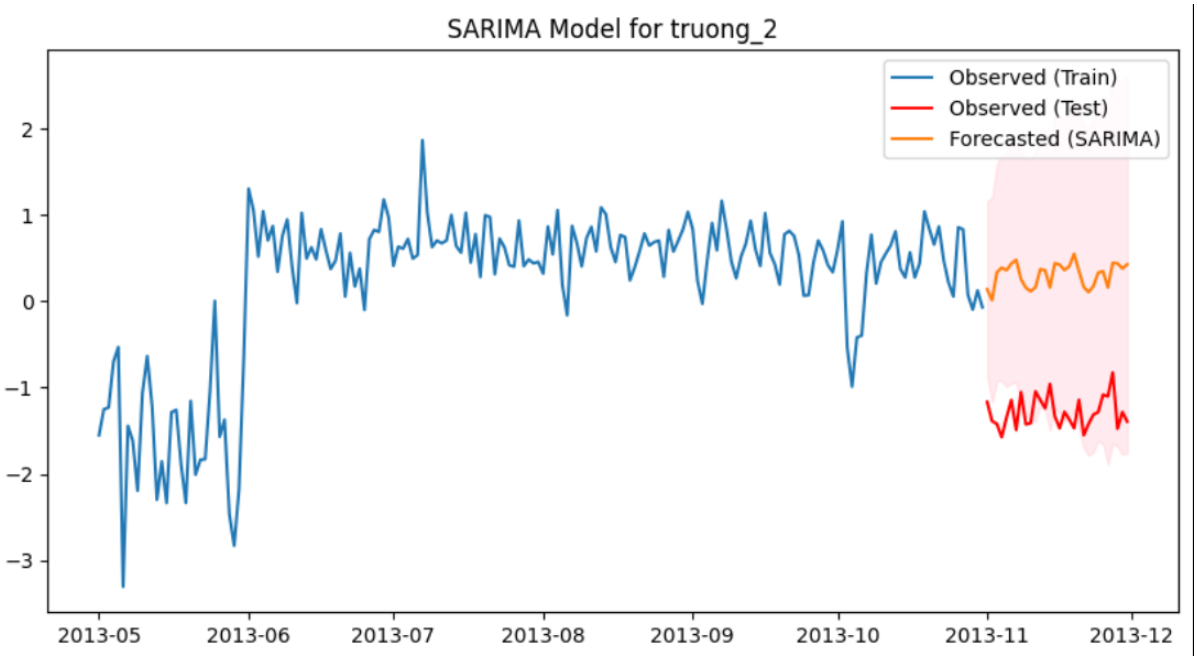
     train = data_filled.iloc[:-30]
     test = data_filled.iloc[-30:]

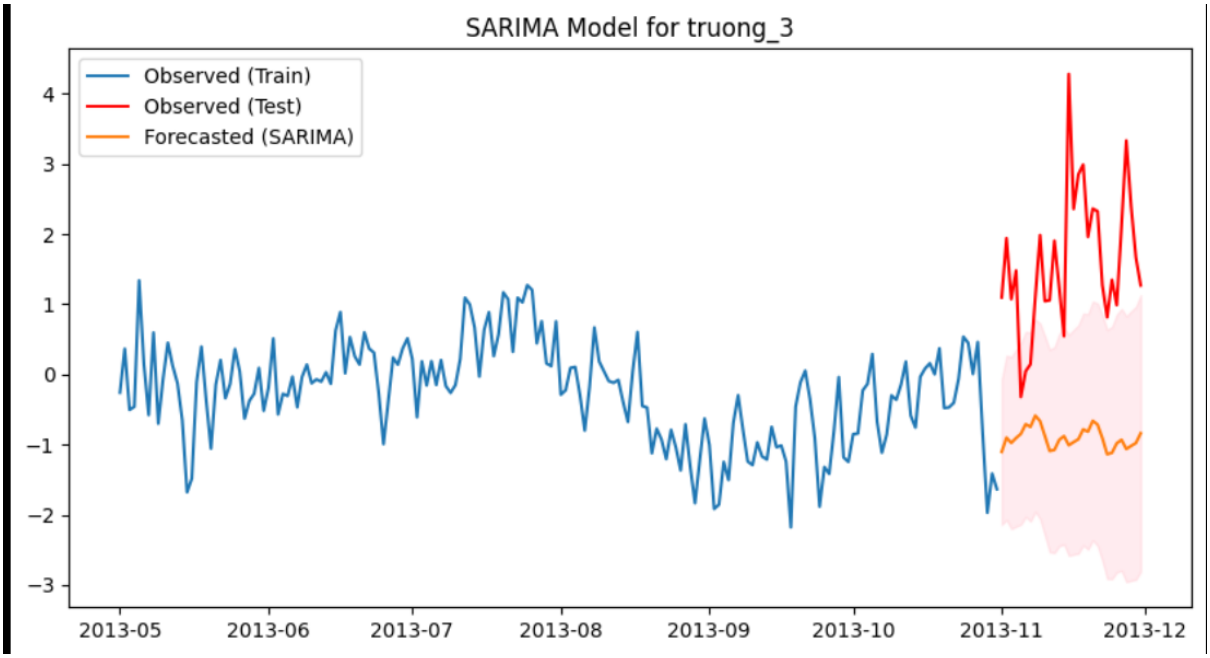
     # Mô hình SARIMA
     Train mô hình với các tham số
     p = 1
     d = 1
     q = 1
     P = 1
     D = 1
     Q = 1
     s = 12
     sarima_model = sm.tsa.statespace.SARIMAX(train[column], order=(1, 1, 1), seasonal_order=(1, 1, 1, 12))
     sarima_result = sarima_model.fit()
     # Dự báo với mô hình SARIMA
     Dự báo với số bước chạy là 30
     sarima_forecast = sarima_result.get_forecast(steps=30)
     sarima_predicted_mean = sarima_forecast.predicted_mean
     sarima_conf_int = sarima_forecast.conf_int()

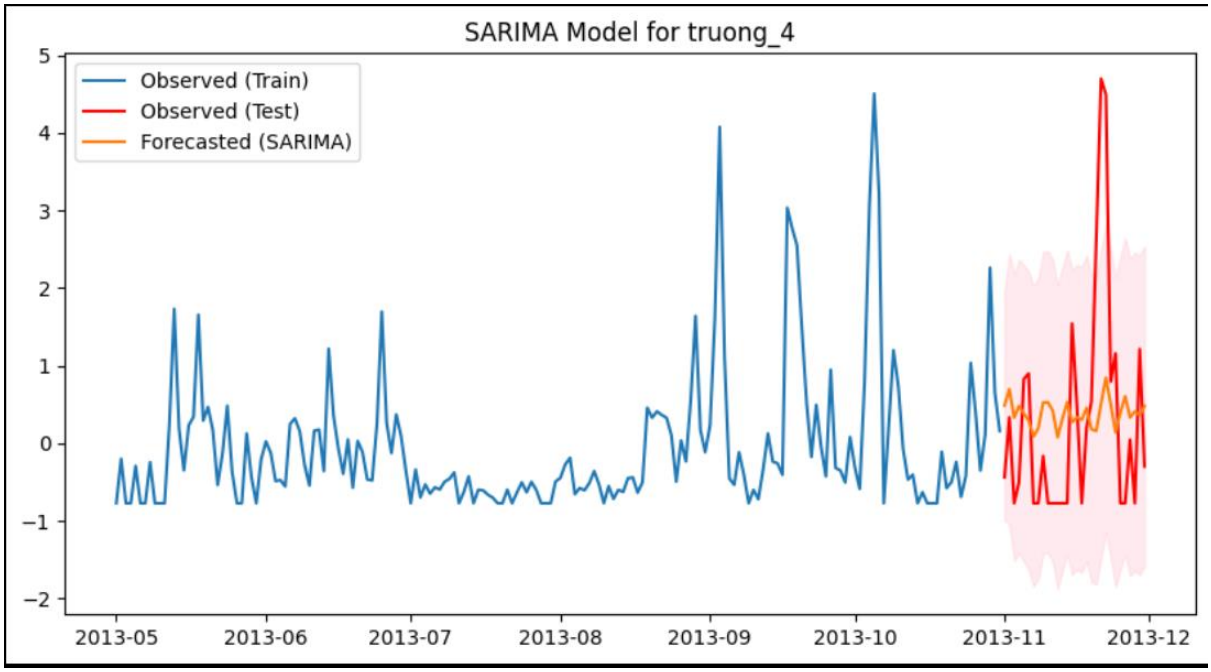
```
# Vẽ đồ thị kết quả dự báo SARIMA
    plt.figure(figsize=(10, 5))
    plt.plot(train.index, train[column], label='Observed (Train)')
    plt.plot(test.index, test[column], label='Observed (Test)', color='red')
    plt.plot(test.index, sarima_predicted_mean, label='Forecasted
(SARIMA)')
    plt.fill_between(test.index, sarima_conf_int.iloc[:, 0],
sarima_conf_int.iloc[:, 1], color='pink', alpha=0.3)
    plt.legend()
    plt.title(f'SARIMA Model for {column}')
    plt.show()
```

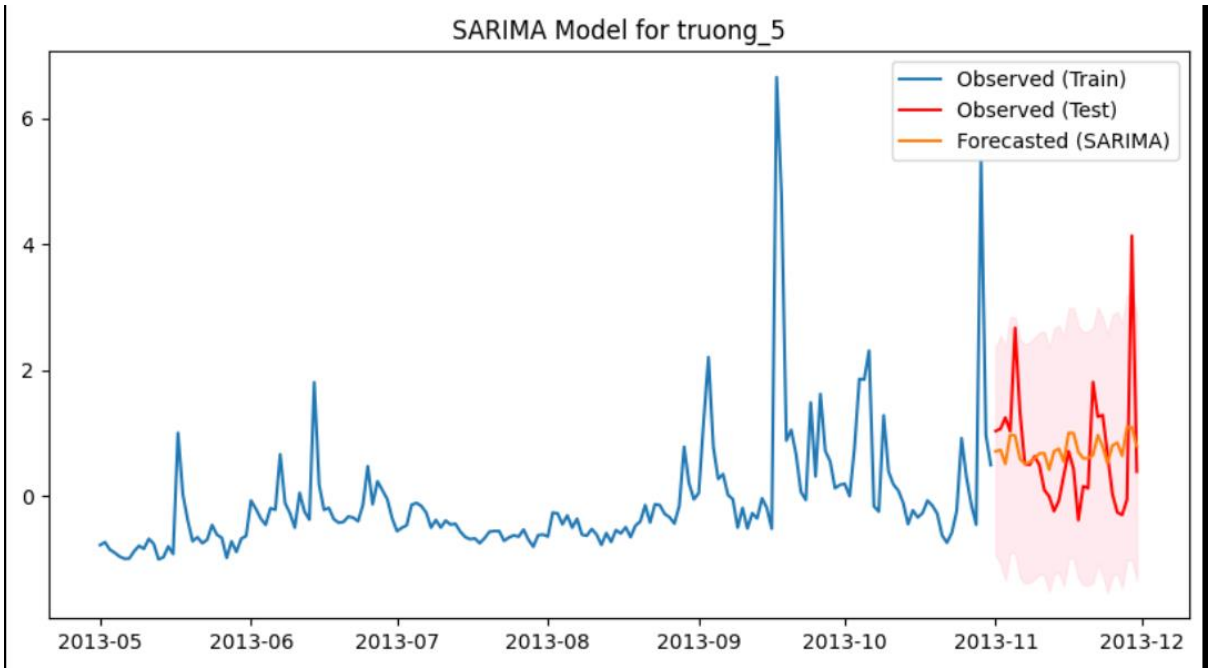
Ta được kết quả:




SARIMA Model for truong_2
Observed (Train)
Observed (Test)
Forecasted (SARIMA)
2
1
0
−1
−2
−3
2013-05
2013-06
2013-07
2013-08
2013-09
2013-10
2013-11
2013-12


SARIMA Model for truong_3
Observed (Train)
Observed (Test)
Forecasted (SARIMA)
4
3
2
1
0
−1
−2
−3
2013-05
2013-06
2013-07
2013-08
2013-09
2013-10
2013-11
2013-12





Mô hình ARIMAX

Train mô hình với các tham số

p = 1

d = 1

q = 1

P = 1

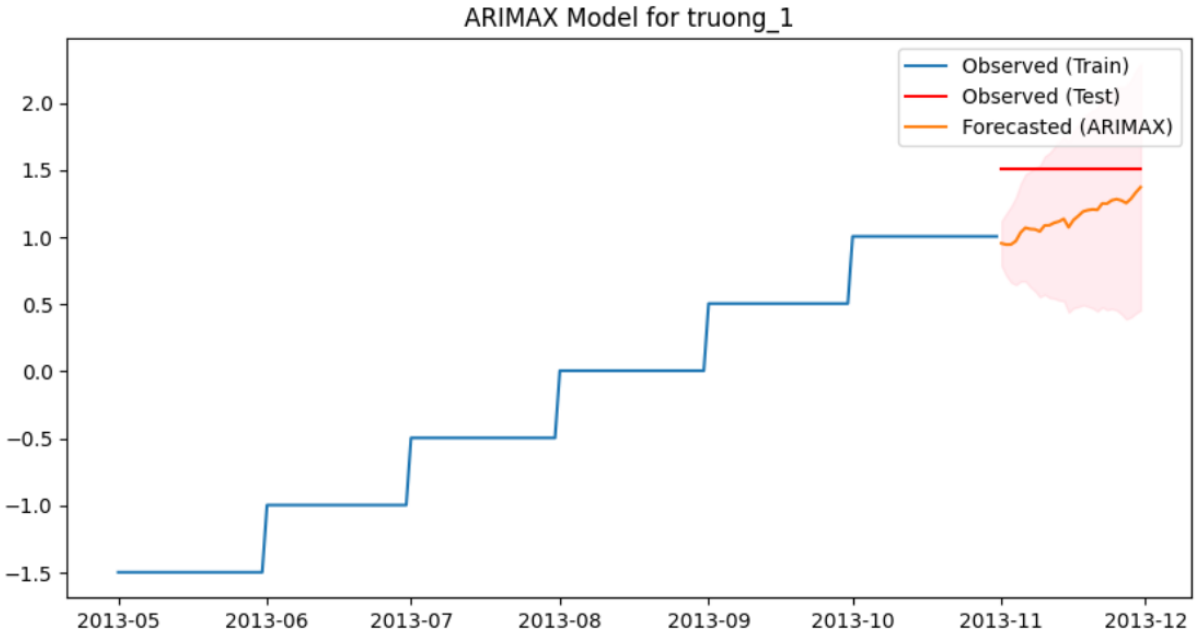
D = 1

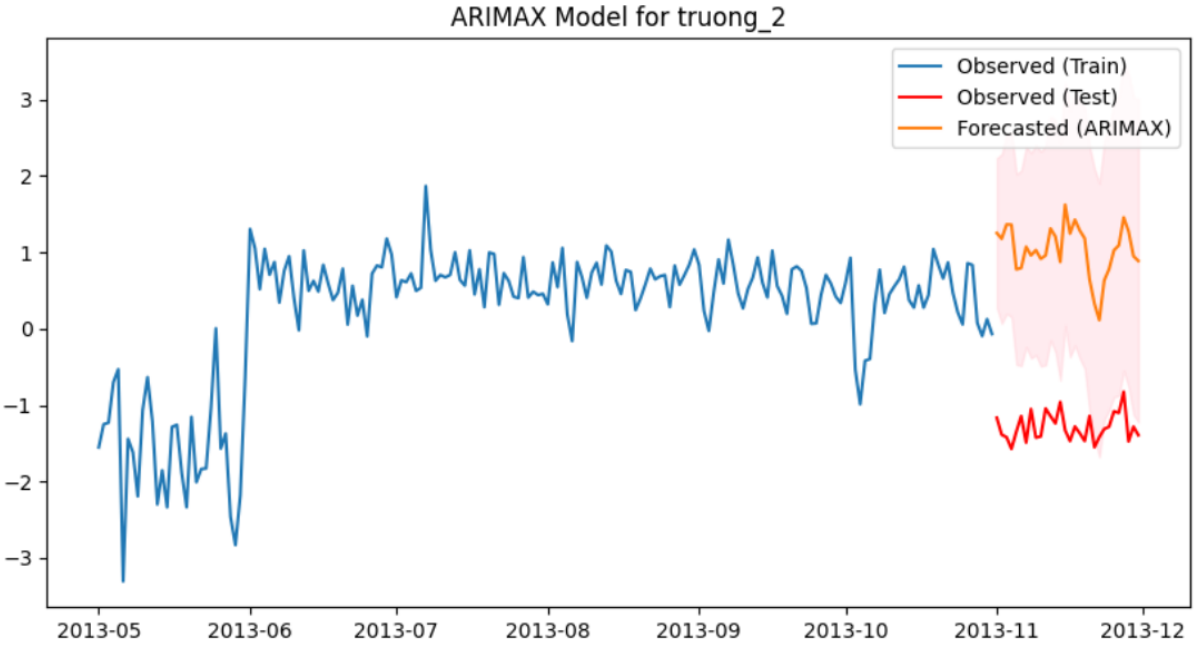
Q = 1

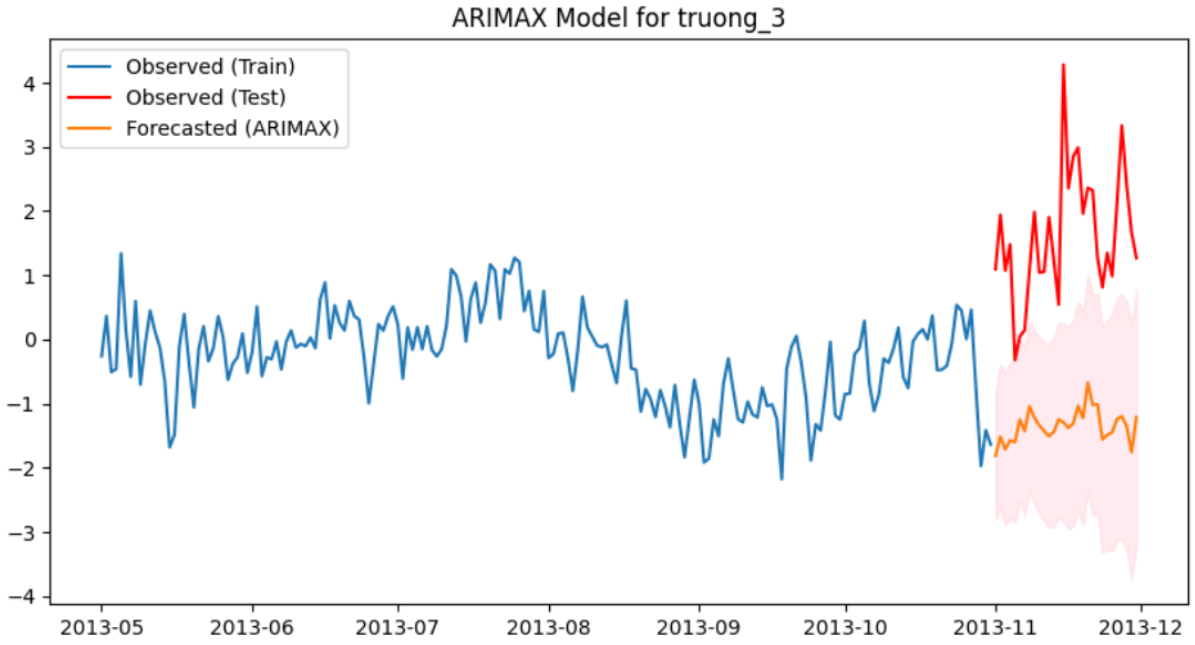
s = 12

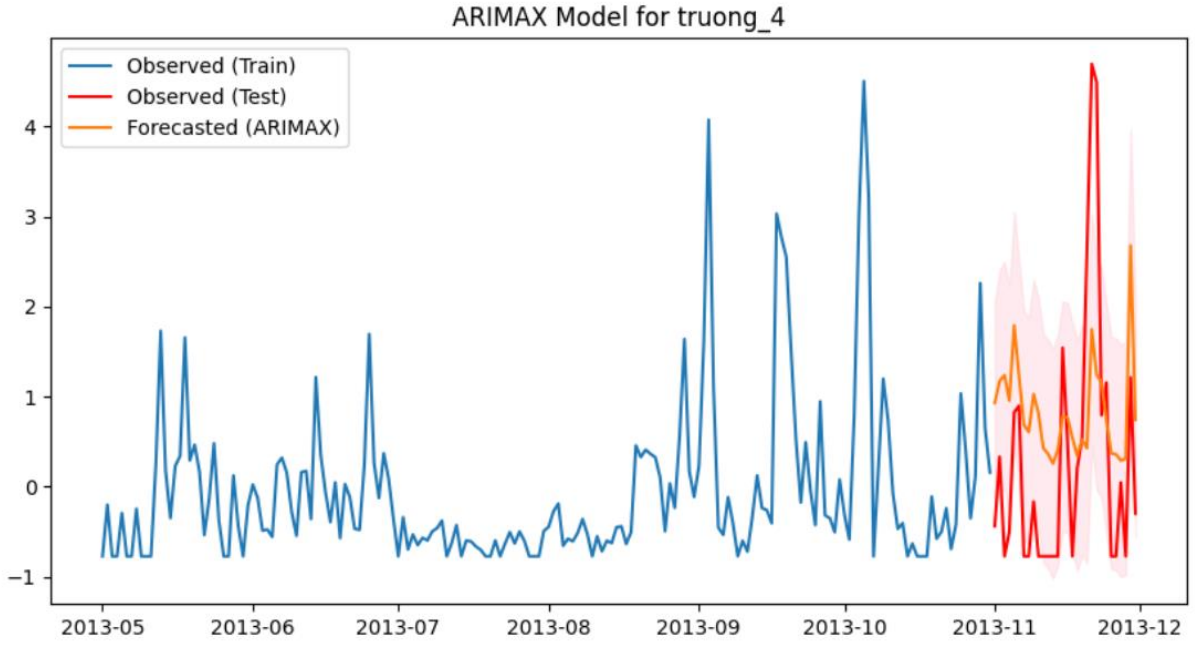
```
    exog_train = train.drop(columns=[column])
    exog_test = test.drop(columns=[column])

    arimax_model = sm.tsa.statespace.SARIMAX(train[column], order=(1,
1, 1), seasonal_order=(1, 1, 1, 12), exog=exog_train)
    arimax_result = arimax_model.fit()

    # Dự báo với mô hình ARIMAX
     Dự báo với số bước chạy là 30
    arimax_forecast = arimax_result.get_forecast(steps=30,
exog=exog_test)
    arimax_predicted_mean = arimax_forecast.predicted_mean
    arimax_conf_int = arimax_forecast.conf_int()
    arimax_forecasts[column] = arimax_predicted_mean

    # Vẽ đồ thị kết quả dự báo ARIMAX
    plt.figure(figsize=(10, 5))
    plt.plot(train.index, train[column], label='Observed (Train)')
    plt.plot(test.index, test[column], label='Observed (Test)', color='red')
    plt.plot(test.index, arimax_predicted_mean, label='Forecasted
(ARIMAX)')
    plt.fill_between(test.index, arimax_conf_int.iloc[:, 0],
arimax_conf_int.iloc[:, 1], color='pink', alpha=0.3)
    plt.legend()
    plt.title(f'ARIMAX Model for {column}')
    plt.show()
```

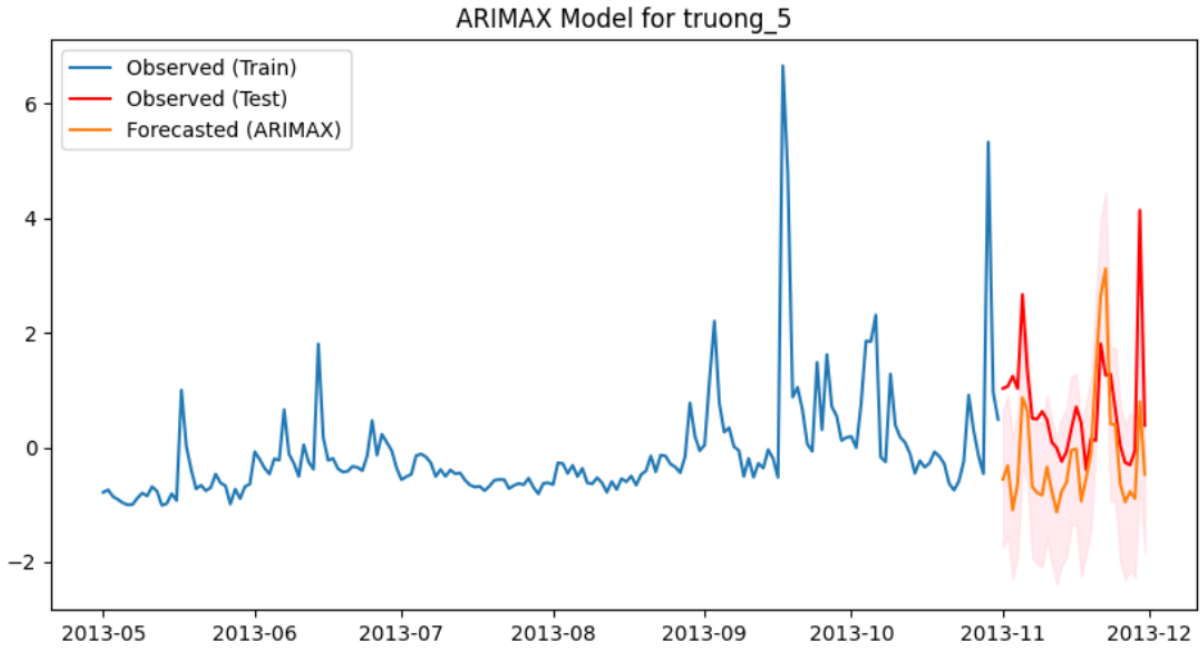
Ta được kết quả:


ARIMAX Model for truong_1
Observed (Train)
Observed (Test)
Forecasted (ARIMAX)
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
2013-05
2013-06
2013-07
2013-08
2013-09
2013-10
2013-11
2013-12


ARIMAX Model for truong_2
Observed (Train)
Observed (Test)
Forecasted (ARIMAX)
3
2
1
0
−1
−2
−3
2013-05
2013-06
2013-07
2013-08
2013-09
2013-10
2013-11
2013-12


ARIMAX Model for truong_3
Observed (Train)
Observed (Test)
Forecasted (ARIMAX)
4
3
2
1
0
−1
−2
−3
−4
2013-05
2013-06
2013-07
2013-08
2013-09
2013-10
2013-11
2013-12


ARIMAX Model for truong_4
Observed (Train)
Observed (Test)
Forecasted (ARIMAX)
4
3
2
1
0
−1
2013-05
2013-06
2013-07
2013-08
2013-09
2013-10
2013-11
2013-12


ARIMAX Model for truong_5
Observed (Train)
Observed (Test)
Forecasted (ARIMAX)
6
4
2
0
−2
2013-05
2013-06
2013-07
2013-08
2013-09
2013-10
2013-11
2013-12