Họ và tên: Tống Văn Lực

MSV: 2151264670

Mô hình Arima và Garch

- Mô hình Arima

ARIMA là phương pháp dự báo yếu tố nghiên cứu một cách độc lập (dự báo theo chuỗi thời gian). Bằng các thuật toán sử dụng độ trễ sẽ đưa ra mô hình dự báo thích hợp.ARIMA là mô hình là kết hợp của 2 mô hình tự hồi quy và trung bình trượt, dữ liệu ở quá khứ được sử sử dụng để dự báo cho tương lai.

Công thức tổng quát của mô hình:

$$Yt=c+\phi 1Yt-1+\phi 2Yt-2+...+\phi pYt-p+\epsilon t+\theta 1\epsilon t-1+\theta 2\epsilon t-2+...+\theta q\epsilon t-q$$

Trong đó:

- Yt: là giá trị hiện tại.
- Φi: là hệ số tự hồi quy.
- Θi: là hệ số trung bình động.
- ϵt: là sai số.

Do phương pháp Box-Jenkins chỉ mô tả chuỗi dừng hoặc những chuỗi đã sai phân hóa, nên mô hình ARIMA(p,d,q) thể hiện những chuỗi dữ liệu không dừng đã được lấy sai phân bậc d. Khi chuỗi thời gian dừng được lựa chọn (hàm tự tương quan ACF giảm đột ngột hoặc giảm đều nhanh), chúng ta có thể chỉ ra một mô hình dự định bằng cách nghiên cứu xu hướng của hàm tự tương quan ACF và hàm tự tương quan từng phần PACF

- Mô hình Garch

Mô hình GARCH có khả năng mô hình hóa sự biến động thay đổi theo thời gian, cho phép dự báo tốt hơn về rủi ro và sự biến động trong các thị trường tài chính. Nó cũng cho phép mô hình hóa các chuỗi thời gian có tính chất "cụm" của biến động, tức là các giai đoạn có biến động cao và thấp thường xuất hiện liên tiếp.

Mô hình GARCH được sử dụng rộng rãi trong các lĩnh vực tài chính như quản lý rủi ro, dự báo biến động giá cổ phiếu, xác định giá trị của các công cụ tài chính phái sinh, và trong các lĩnh vực khác như kinh tế vĩ mô và dự báo nhu cầu năng lượng.

Mô hình GARCH(p,q) có thể được biểu diễn qua hai phương trình chính:

Phương trình hồi quy:

$$Y_t = \mu + \epsilon_t$$

Phương trình phương sai có điều kiện:

$$\sigma_t^2 = \alpha_0 + \alpha_1\epsilon_{t-1}^2 + \alpha_2\epsilon_{t-2}^2 + ... + \alpha_p\epsilon_{t-p}^2 + \beta_1\sigma_{t-1}^2 + \beta_2\sigma_{t-2}^2 + ... + \beta_q\sigma_{t-q}^2$$

Đầu tiên sẽ tiền xử lý dữ liệu bằng cách lấy trung bình những ngày bị trùng lặp.

```
# df['date'] = pd.to_datetime(df['date'], format='%m/%d/%Y')
df_gop = df.groupby('date').mean().reset_index()
len(df_gop['date'])

# lưu data gộp vào file data_kiem_tra_gop.csv
# df_gop.to_csv('data-kiem-tra-gop.csv', index=False)
```

Executed at 2024.06.04 12:14:21 in 46ms

```
214
```

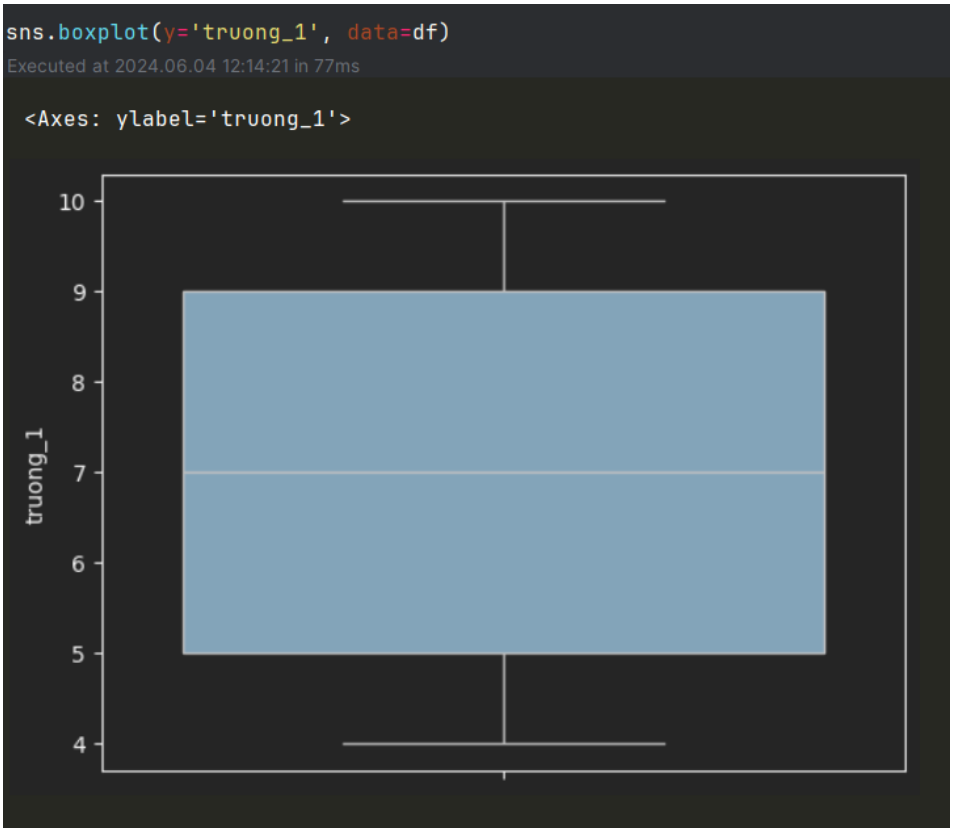
Sau đó xử lý giá trị ngoại lai của các trường

Trường 1:

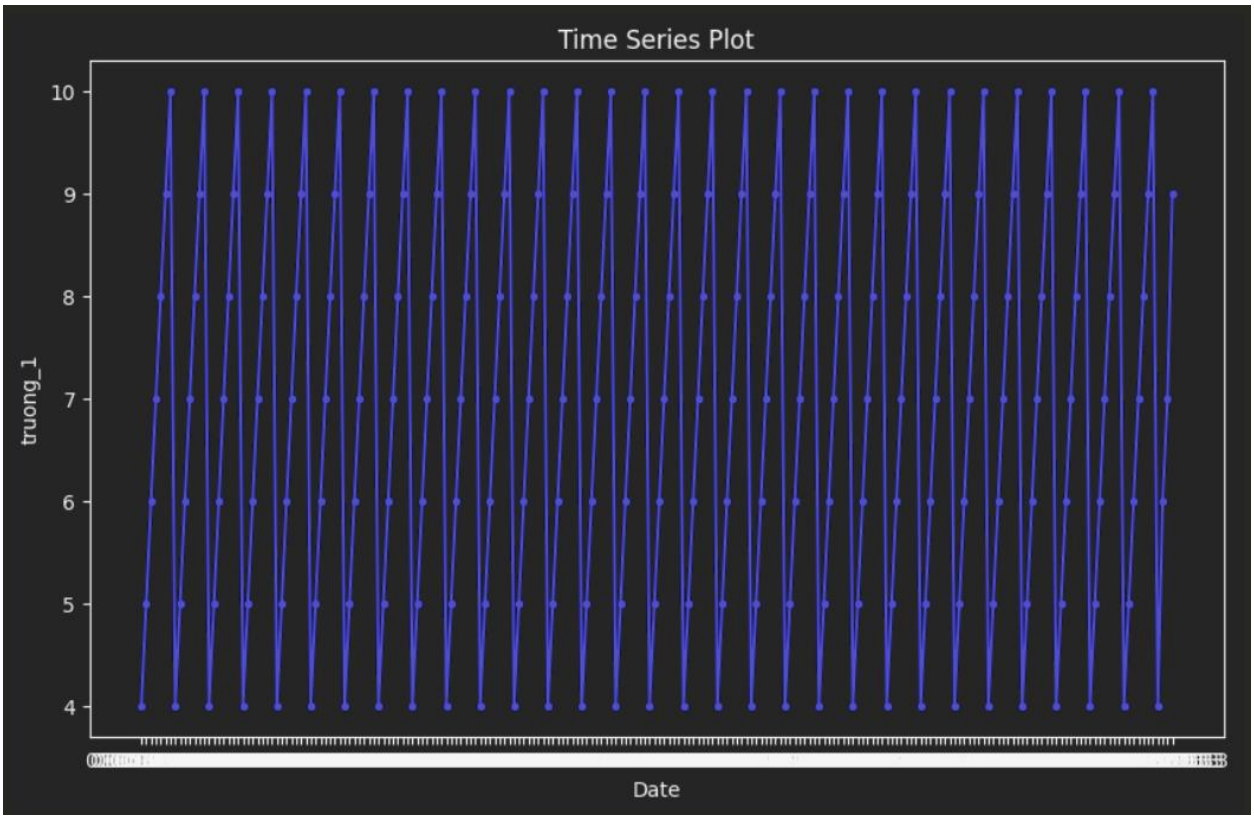
```
sns.boxplot(y='truong_1', data=df)
```

Executed at 2024.06.04 12:14:21 in 77ms

```
<Axes: ylabel='truong_1'>
```



Vì không có giá trị ngoại lai nên vẽ luôn biểu đồ timeseries với trường 1:



Làm tương tự với các cột còn lại, nếu có giá trị ngoại lai thì xử lý bằng cách đưa giá trị lớn hơn max về max, giá trị nhỏ hơn min về min.

Mô hình ARIMA:

Ta dùng code lựa chọn ra 3 tham số tốt nhất là (1, 0, 2):

```
p = range(0, 3)  # Số lượng lag cho phần autoregressive (AR)
d = range(0, 2)  # Số lượng lần sai phân
q = range(0, 3)  # Số lượng lag cho phần moving average (MA)

# Tạo danh sách tất cả các tổ hợp của p, d, q
pdq = list(itertools.product(p, d, q))

# Lặp qua tất cả các tổ hợp để tìm ra bộ tham số tốt nhất
best_score, best_params = float("inf"), None
for param in pdq:
    try:
        model = ARIMA(df['truong_3'], order=param)
        result = model.fit()
        # Đánh giá mô hình sử dụng AIC
        if result.aic < best_score:
            best_score, best_params = result.aic, param
    except:
        continue

print("Best AIC:", best_score)
print("Best Parameters:", best_params)
```

Executed at 2024.06.04 12:14:27 in 865ms

```
Best AIC: 3168.308837798416
Best Parameters: (1, 0, 2)
```

Sau đó đưa tham số để chạy mô hình và dự đoán ra 15 ngày trong tương lai:

-train mô hình:

```
model = ARIMA(df['truong_3'], order=best_params)
result = model.fit()
result.summary()
```

Executed at 2024.06.04 12:14:28 in 139ms

SARIMAX Results

Dep. Variable:	truong_3	**No. Observations:**	214
Model:	ARIMA(1, 0, 2)	**Log Likelihood**	-1579.154
Date:	Tue, 04 Jun 2024	**AIC**	3168.309
Time:	12:14:28	**BIC**	3185.139
Sample:	0	**HQIC**	3175.110
	- 214		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
const	1.038e+04	19.508	532.182	0.000	1.03e+04	1.04e+04
ar.L1	-0.6355	0.084	-7.593	0.000	-0.800	-0.471
ma.L1	0.5539	0.094	5.899	0.000	0.370	0.738
ma.L2	-0.3859	0.076	-5.062	0.000	-0.535	-0.236
sigma2	1.495e+05	1.45e+04	10.306	0.000	1.21e+05	1.78e+05

Ljung-Box (L1) (Q):	0.23	**Jarque-Bera (JB):**	0.02
Prob(Q):	0.63	**Prob(JB):**	0.99
Heteroskedasticity (H):	1.79	**Skew:**	-0.02
Prob(H) (two-sided):	0.02	**Kurtosis:**	3.00

-dự đoán ra 15 ngày trong tương lai:

```
forecast_values = result.forecast(steps=15)
forecast_values
```

Executed at 2024.06.04 12:14:28 in 8ms

1-10 Length: 15, dtype: float64

	predicted_mean
214	10797.688009
215	10281.441243
216	10445.302914
217	10341.171878

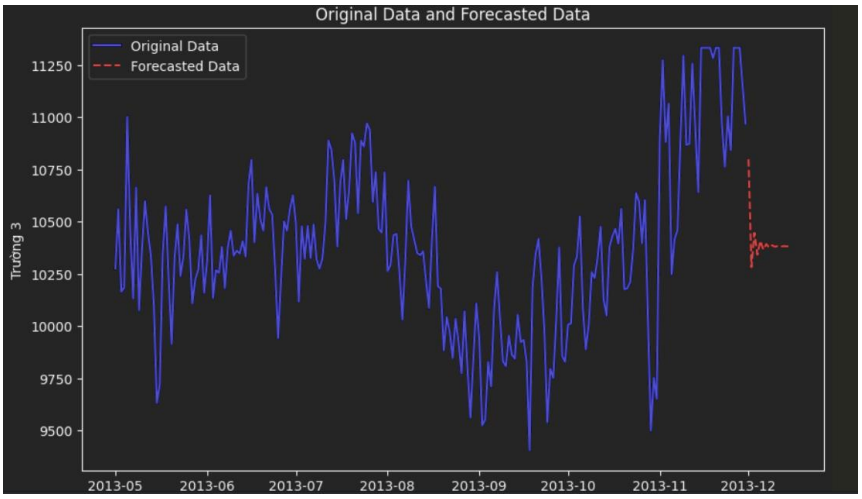
Biểu đồ biểu diễn:

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

df = df.sort_values(by='date')

# Vẽ biểu đồ
plt.figure(figsize=(10, 6))
plt.plot(df['date'], df['truong_3'], color='blue', label='Original Data')
plt.plot(pd.date_range(start=df['date'].iloc[-1], periods=16)[1:], forecast_values, color='red', linestyle='--', label='Forecasted Data')
plt.xlabel('Date')
plt.ylabel('Trường 3')
plt.title('Original Data and Forecasted Data')
plt.legend()
plt.show()
```

Executed at 2024.06.04 12:14:28 in 147ms



Mô hình GARCH:

Tương tự mô hình ARIMA ta cũng dùng code để lựa chọn tham số p và q cho mô hình:

```
# Khởi tạo biến lưu trữ kết quả tốt nhất và tham số tương ứng
best_aic = np.inf
best_params = None

# Lặp qua tất cả các tổ hợp của tham số p và q
for params in ParameterGrid(params_grid):
    try:
        # Xây dựng mô hình GARCH với các tham số đã chọn
        model = arch_model(df['truong_3'], vol='GARCH', p=params['p'], q=params['q'])
        result = model.fit(disp='off')

        # Lưu kết quả tốt nhất
        if result.aic < best_aic:
            best_aic = result.aic
            best_params = params
    except:
        continue

print("Best AIC:", best_aic)
print("Best Parameters:", best_params)
```

Executed at 2024.06.04 12:15:24 in 20s 870ms

```
C:\Users\ASUS\AppData\Local\Programs\Python\Python312\Lib\site-packages\arch\univariate\base.py:311: DataScaleWarning: y

Best AIC: 3105.6995737631896
Best Parameters: {'p': 1, 'q': 2}
```

Như ta thấy 2 tham số tốt nhất là p=1 và q=2

Đưa tham số trên và để train mô hình:

```
model = arch_model(df['truong_3'], vol='GARCH', p=best_params['p'], q=best_params['q'])
result = model.fit(disp='off')
result.summary()
```

Executed at 2024.06.04 12:15:31 in 81ms

```
C:\Users\ASUS\AppData\Local\Programs\Python\Python312\Lib\site-packages\arch\univariate\
```

Constant Mean - GARCH Model Results

Dep. Variable:	truong_3	**R-squared:**	0.000
Mean Model:	Constant Mean	**Adj. R-squared:**	0.000
Vol Model:	GARCH	**Log-Likelihood:**	-1547.85
Distribution:	Normal	**AIC:**	3105.70
Method:	Maximum Likelihood	**BIC:**	3122.53
		No. Observations:	214
Date:	Tue, Jun 04 2024	**Df Residuals:**	213
Time:	12:15:31	**Df Model:**	1

Mean Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
mu	1.0371e+04	23.658	438.380	0.000	[1.032e+04,1.042e+04]

Volatility Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
omega	1.6713e+04	8665.954	1.929	5.378e-02	[-2.715e+02,3.370e+04]
alpha[1]	0.6463	0.196	3.292	9.950e-04	[0.262, 1.031]

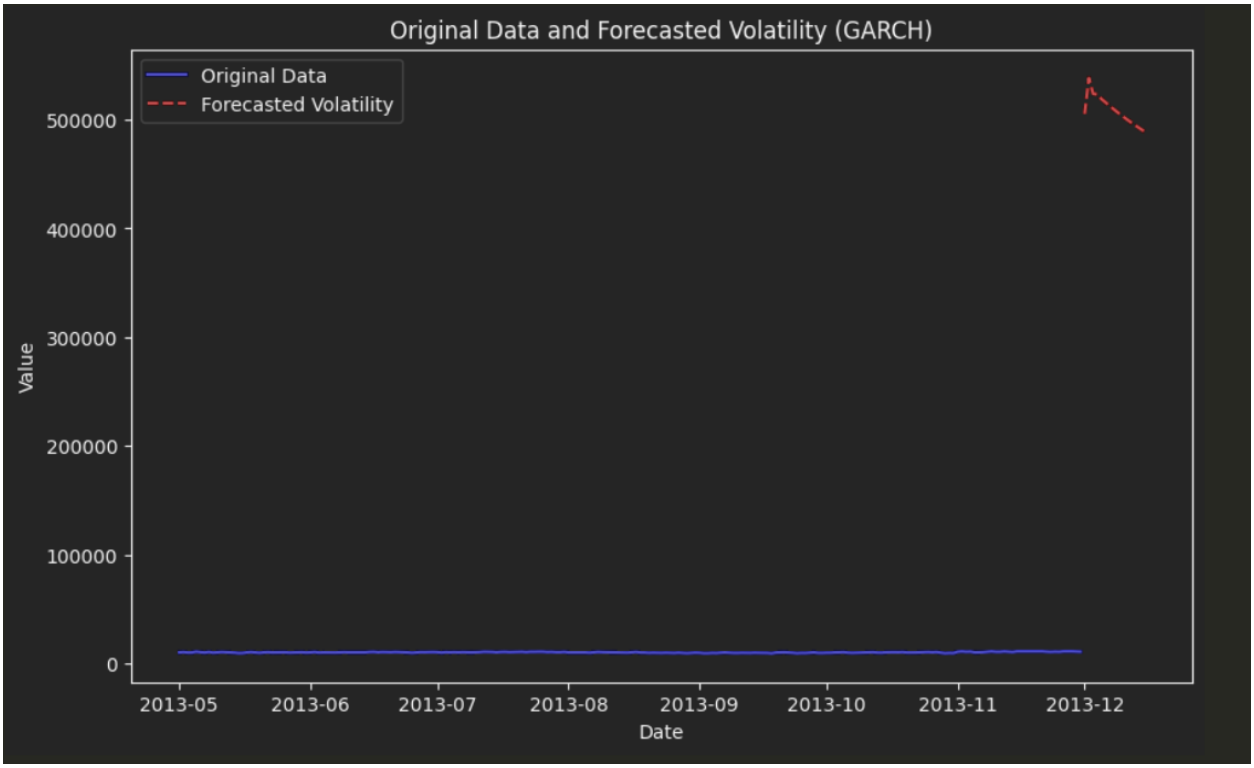
Ta cũng dự đoán ra 15 ngày vẽ biểu đồ biểu diễn.

```
forecast_values = result.forecast(horizon=15)
forecast_values = forecast_values
```

Executed at 2024.06.04 12:15:34 in 19ms

```
# Vẽ biểu đồ
plt.figure(figsize=(10, 6))
plt.plot(df['date'], df['truong_3'], color='blue', label='Original Data')
plt.plot(pd.date_range(start=df['date'].iloc[-1], periods=16)[1:], forecast_values.variance.iloc[-1, :], color='red', linestyle='--', label='Forecasted Volatility')
plt.xlabel('Date')
plt.ylabel('Value')
plt.title('Original Data and Forecasted Volatility (GARCH)')
plt.legend()
plt.show()
```

Executed at 2024.06.04 12:15:36 in 166ms



Theo 2 biểu đồ trên ta thấy mô hình ARIMA dự đoán khá tốt và sai số có vẻ thấp so với giá trị thực tế, trái ngược lại đó thì ta thấy mô hình GARCH dự đoán sai số với giá trị thực tế rất cao.