

Link github:

1. Bài 1HMM

Kiểm tra tham số của các mô hình:



```
import numpy as np
from hmmlearn import hmm

# Chuẩn bị dữ liệu cho HMM
data = df[['12', '19', '21']].values

# Xây dựng mô hình HMM với GaussianHMM
model = hmm.GaussianHMM(n_components=3, covariance_type='diag', n_iter=1000)
model.fit(data)

# Kiểm tra các tham số của mô hình
print("Means and vars of each hidden state")
for i in range(model.n_components):
    print(f"State {i}:")
    print("Mean =", model.means_[i])
    print("Var =", np.diag(model.covars_[i]))
```

Means and vars of each hidden state
State 0:
Mean = [987.13015539 987.10469245 985.30839765]
Var = [859.29428514 866.39243642 1250.2094233]
State 1:
Mean = [1388.98750074 1388.84166741 1381.00416743]
Var = [9601.47797793 9604.27181495 11399.0181057]
State 2:
Mean = [1065.26320467 1065.26880467 1065.26320467]
Var = [14.24352599 14.21390681 14.24352576]

State 0: Có giá trị trung bình tương đối cao và phương sai lớn, cho thấy sự biến động lớn trong trạng thái này. Điều này có thể ám chỉ rằng trạng thái này xảy ra trong các giai đoạn có nhiều biến động trên thị trường.

State 1: Có giá trị trung bình cao nhất và phương sai rất lớn, cho thấy sự biến động cực kỳ cao. Trạng thái này có thể đại diện cho các giai đoạn thị trường có các biến động rất mạnh hoặc bất thường.

State 2: Có giá trị trung bình ở mức trung bình và phương sai rất nhỏ, cho thấy sự ổn định trong trạng thái này. Trạng thái này có thể đại diện cho các giai đoạn thị trường ổn định, ít biến động.

Vẽ biểu đồ xem phân bố các trạng thái ẩn của cột 12:

```

# Dự đoán các trạng thái ẩn cho dữ liệu
hidden_states = model.predict(data)

# Thêm các trạng thái ẩn vào dataframe
df['Hidden State'] = hidden_states

# Hiển thị kết quả
print(df.head())

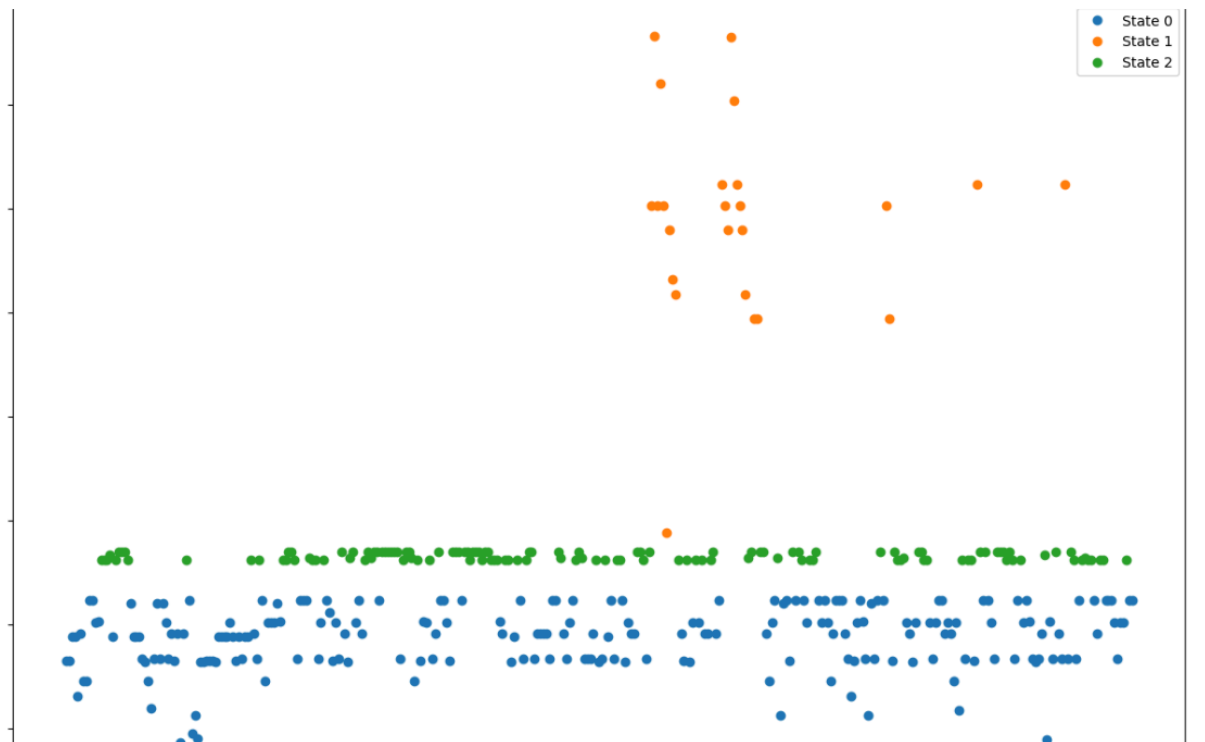
# Vẽ biểu đồ để xem phân bố các trạng thái ẩn
import matplotlib.pyplot as plt

plt.figure(figsize=(15, 10))

for i in range(model.n_components):
    state = (hidden_states == i)
    plt.plot(df.index[state], df['12'][state], 'o', label=f'State {i}')

plt.legend()
plt.show()

```



1. State 0 (màu xanh dương):

- Các điểm dữ liệu nằm khá gần nhau và có xu hướng tập trung ở phần dưới của biểu đồ.
- Trạng thái này có giá trị trung bình thấp nhất và phương sai trung bình.
- Điều này cho thấy State 0 đại diện cho các giai đoạn giá trị SMP tương đối thấp và có biến động vừa phải.

2. State 1 (màu cam):

- Các điểm dữ liệu nằm rải rác hơn và tập trung ở phần trên của biểu đồ.
- Trạng thái này có giá trị trung bình cao nhất và phương sai rất lớn.

- Điều này chỉ ra rằng State 1 đại diện cho các giai đoạn giá trị SMP rất cao và có sự biến động rất mạnh.

3. State 2 (màu xanh lá cây):

- Các điểm dữ liệu nằm khá gần nhau và tập trung ở phần giữa của biểu đồ.
- Trạng thái này có giá trị trung bình ở mức trung bình và phương sai nhỏ nhất.
- Điều này cho thấy State 2 đại diện cho các giai đoạn giá trị SMP ổn định và ít biến động.

Phân tích cụ thể

- **Giá trị SMP thấp (State 0):** Khi giá trị SMP thấp, trạng thái ẩn chủ yếu là State 0. Điều này có thể xảy ra trong các giai đoạn thị trường bình ổn hoặc giảm giá.
- **Giá trị SMP cao (State 1):** Khi giá trị SMP cao, trạng thái ẩn chủ yếu là State 1. Điều này có thể phản ánh các giai đoạn thị trường có sự tăng trưởng mạnh hoặc có những biến động lớn.
- **Giá trị SMP ổn định (State 2):** Khi giá trị SMP ổn định, trạng thái ẩn chủ yếu là State 2. Điều này chỉ ra các giai đoạn thị trường ổn định mà không có sự biến động lớn.

Dùng Hmm để dự đoán 30 ngày tiếp theo dựa trên 365 ngày trong dữ liệu:

```
# Khởi tạo mô hình HMM
model = GaussianHMM(n_components=num_states, covariance_type="diag", n_iter=1000)

# Huấn luyện mô hình HMM
model.fit(prices)

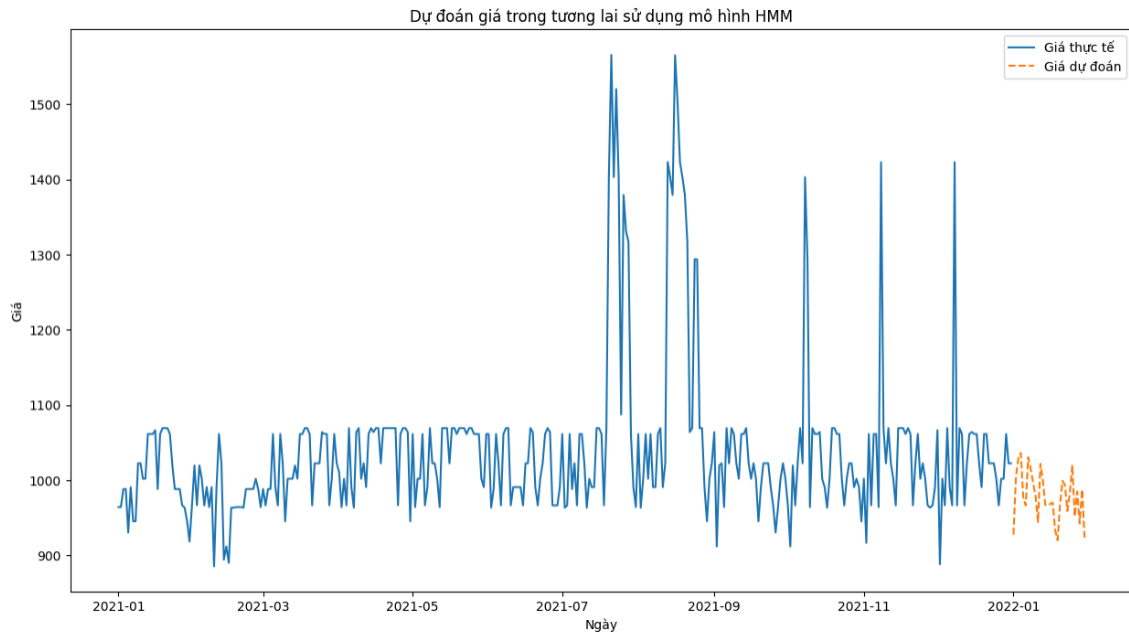
# Lấy các trạng thái ẩn dự đoán
hidden_states = model.predict(prices)

# Sử dụng mô hình đã huấn luyện để dự đoán giá trị trong tương lai
num_days = 30 # Số ngày cần dự đoán
future_prices = []

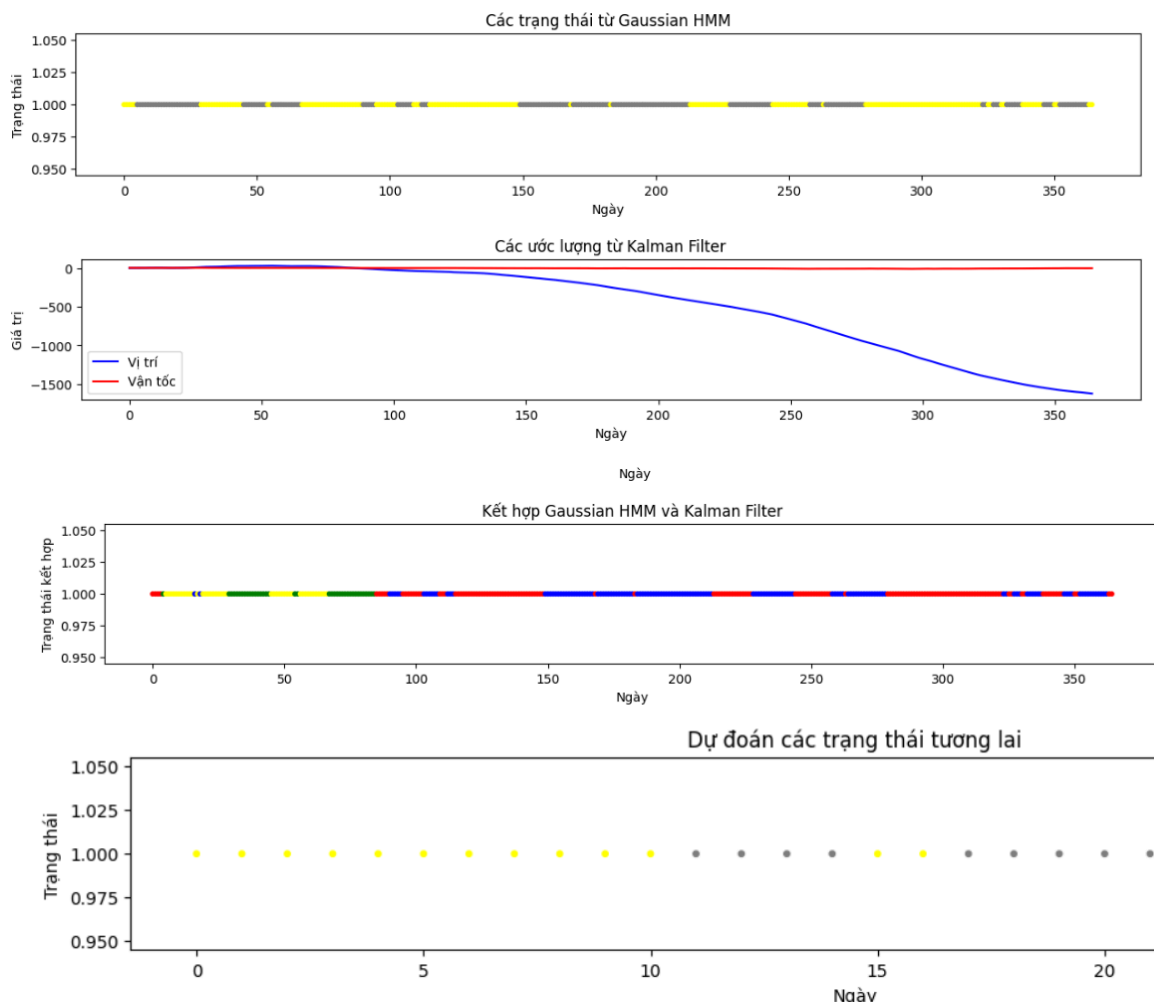
# Sử dụng giá trị cuối cùng để bắt đầu dự đoán
current_price = prices[-1]

for _ in range(num_days):
    # Lấy trạng thái hiện tại
    current_state = model.predict(current_price.reshape(-1, 1))
    # Tính giá trị dự đoán tiếp theo
    next_price = model.means_[current_state] + np.random.normal(0, np.sqrt(model.covars_[current_state])).
    future_prices.append(next_price[0][0])
    current_price = next_price

# Chuyển đổi kết quả thành DataFrame
future_dates = pd.date_range(start=dates[-1] + pd.Timedelta(days=1), periods=num_days)
future_df = pd.DataFrame({'Ngày': future_dates, 'Dự đoán giá': future_prices})
```



• Bài 2 (Dùng GaussianHmm)



Biểu đồ 1: Các trạng thái từ Gaussian HMM

- Trục x: Biểu diễn số ngày (từ 0 đến 365).
- Trục y: Biểu diễn trạng thái thời tiết (chỉ số 1 và 2 tương ứng với các trạng thái thời tiết khác nhau).
- Nội dung: Biểu đồ này cho thấy các trạng thái thời tiết được dự đoán bởi mô hình Gaussian HMM qua các ngày trong năm. Các điểm màu khác nhau biểu thị cho các trạng thái thời tiết khác nhau.

Biểu đồ 2: Các ước lượng từ Kalman Filter

- Trục x: Biểu diễn số ngày (từ 0 đến 365).
- Trục y: Giá trị (các ước lượng từ Kalman Filter).
- Nội dung: Biểu đồ này thể hiện các ước lượng vị trí và vận tốc được tính toán bởi bộ lọc Kalman. Đường màu xanh biểu diễn vị trí và đường màu đỏ biểu diễn vận tốc. Vị trí và vận tốc được dự đoán thay đổi qua thời gian.

Biểu đồ 3: Kết hợp Gaussian HMM và Kalman Filter

- Trục x: Biểu diễn số ngày (từ 0 đến 365).
- Trục y: Trạng thái kết hợp (chỉ số 1 và 2 tương ứng với các trạng thái thời tiết khác nhau).
- Nội dung: Biểu đồ này thể hiện kết quả dự đoán trạng thái thời tiết sau khi đã được xử lý qua bộ lọc Kalman. Các trạng thái thời tiết được làm mịn và dự đoán chính xác hơn so với biểu đồ 1.

Tóm tắt

Các biểu đồ trên giúp chúng ta hiểu rõ hơn về cách mô hình Gaussian HMM dự đoán các trạng thái thời tiết hàng ngày và cách bộ lọc Kalman cải thiện các dự đoán này bằng cách làm mịn dữ liệu. Điều này giúp cải thiện độ chính xác và độ tin cậy của dự đoán thời tiết cho 30 ngày tiếp theo.