

Kalman:

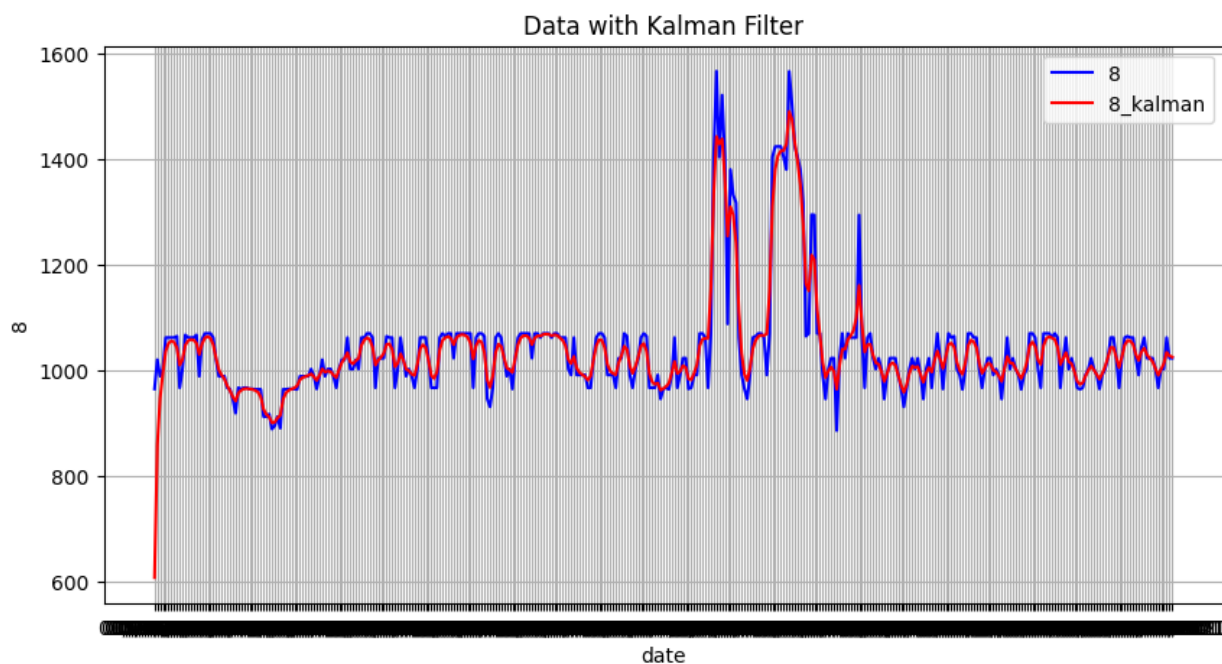
Data được lọc bằng Kalman filter:

```
[18] from pykalman import KalmanFilter
      # Khởi tạo bộ lọc Kalman
      kf = KalmanFilter(initial_state_mean=0, n_dim_obs=1)

      # Áp dụng bộ lọc Kalman cho từng cột dữ liệu
      df[f'{col1}_kalman'] = kf.smooth(df[str(col1)].values)[0]
      df[f'{col2}_kalman'] = kf.smooth(df[str(col2)].values)[0]
      df[f'{col3}_kalman'] = kf.smooth(df[str(col3)].values)[0]

      # Trực quan hóa kết quả sau khi áp dụng bộ lọc Kalman
      fig, ax = plt.subplots(figsize = (10,5))
      sns.lineplot(ax=ax, data=df, x="date", y=str(col1), label=col1, color='b')
      sns.lineplot(ax=ax, data=df, x="date", y=f'{col1}_kalman', label=f'{col1}_kalman', color='r')
      plt.title('Data with Kalman Filter')
      plt.legend()
      plt.grid(True)
      plt.show()
```

Bộ dữ liệu sau khi lọc:



HMM:

```
[21] # Chọn cột dữ liệu đã được lọc bằng bộ lọc Kalman
kalman_cols = [f'{col1}_kalman', f'{col2}_kalman', f'{col3}_kalman']
kalman_data = df[kalman_cols]

# Khởi tạo mô hình HMM với 3 trạng thái ẩn
model = hmm.GaussianHMM(n_components=3, covariance_type='diag', n_iter=1000)

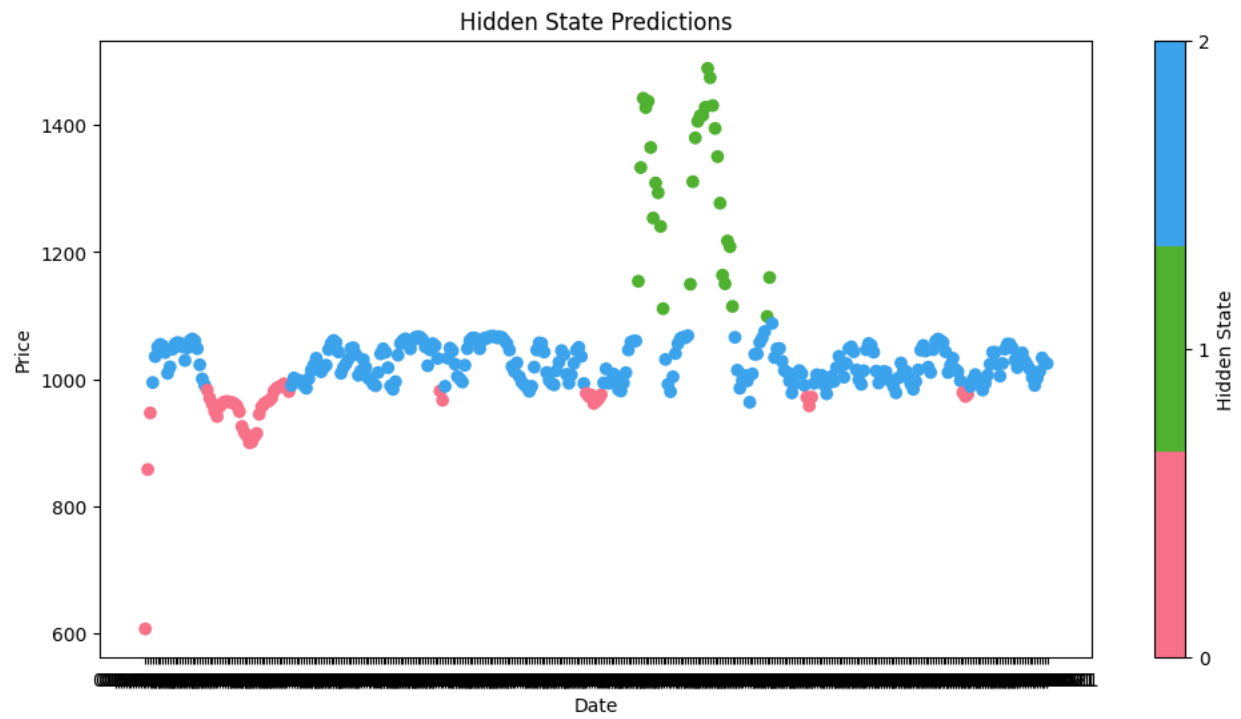
# Huấn luyện mô hình với dữ liệu đã được lọc
model.fit(kalman_data)

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

# Thêm cột trạng thái ẩn vào DataFrame
df['hidden_state'] = hidden_states

# Trực quan hóa kết quả dự đoán
plt.figure(figsize=(12, 6))
cmap = ListedColormap(sns.color_palette("husl", 3))
plt.scatter(df['date'], df[kalman_cols[0]], c=df['hidden_state'], cmap=cmap)
plt.title('Hidden State Predictions')
plt.xlabel('Date')
plt.ylabel('Price')
plt.colorbar(ticks=[0, 1, 2], label='Hidden State')
plt.show()
```

Kết quả dự đoán trạng thái ẩn:



Link Github:

https://github.com/CuongNgD203/TH4_TimeSeries