trong qui trình Xử lý đã đề cập ban đầu, hãy xác định, việc xử lý ngoại lai với IQR có tương đồng với bước: Bước 1: Giảm biến nhiễu và dư thừa bằng sử dụng độ đo hệ số tương quan Pearson tại Pha giảm chiều dữ liệu cho PCA?

1. Việc trình bày phải thực hiện theo quy trình phân tích dữ liệu 8 bước
2. Cần phương pháp xử lý và khắc phục tình trạng bên dưới của dữ liêu:
   2.1 tình trạng thiếu dữ liệu trong tập dữ liệu được giao bằng sử dụng phương pháp trung bình trượt bậc 5
   2.2 Cân chỉnh trung bình tập dữ liệu đầu vào để tính các thành phần chính;
3. Xây dựng mô hình và thực hiện dự báo Theo quy trình 2 pha bao gồm:
   3.1 Pha giảm chiều; Pha này thực hiện theo quy trình hai bước:
   - Bước 1: Giảm biến nhiễu và dư thừa bằng sử dụng độ đo hệ số tương quan Pearson;
   - Bước 2: Thực hiện phương pháp giảm chiều PCA đối với tập dữ liệu của các biến có liên quan và không dư thừa đối với mục đích dự báo của biến gốc;
   3.2 Pha xây dựng mô hình và thực hiện dự báo. Ở Pha Giảm chiều
   3.3 Yêu cầu: Cần xử lý tính dừng của các biến trước khi thực hiện xây dựng mô hình dự báo.
4. Kết hợp ngưỡng tỷ lệ phương sai tích lũy và riêng giá trị để chọn số lượng các thành phần chính cần được giữ lại.

Để trả lời câu hỏi của bạn, tôi sẽ trình bày một cách chi tiết và có cấu trúc theo **quy trình phân tích dữ liệu 8 bước**, tập trung vào việc xác định mối liên hệ giữa xử lý ngoại lai với IQR và bước giảm biến nhiễu/dư thừa bằng hệ số tương quan Pearson trong pha giảm chiều dữ liệu cho PCA. Đồng thời, tôi sẽ đề xuất phương pháp xử lý tình trạng thiếu dữ liệu, cân chỉnh trung bình, xây dựng mô hình dự báo theo quy trình 2 pha, xử lý tính dừng của các biến, và chọn số lượng thành phần chính dựa trên ngưỡng tỷ lệ phương sai tích lũy và giá trị riêng. Tôi sẽ áp dụng ngữ cảnh của tập dữ liệu `VNINDEX_iqr_filled_HSG_Adjust.csv` (đã xử lý ngoại lai với IQR, nội suy thiếu dữ liệu, và thay `HSG` bằng `HSG_log`).

Quy trình phân tích dữ liệu 8 bước

Quy trình phân tích dữ liệu 8 bước thường bao gồm:

1. **Xác định vấn đề**: Hiểu mục tiêu và yêu cầu phân tích.
2. **Thu thập dữ liệu**: Xác định nguồn dữ liệu và thu thập.
3. **Khám phá dữ liệu (EDA)**: Phân tích đặc điểm, phân phối, và chất lượng dữ liệu.
4. **Làm sạch dữ liệu**: Xử lý dữ liệu thiếu, ngoại lai, và lỗi.
5. **Chuẩn bị dữ liệu**: Chuyển đổi, chuẩn hóa, và giảm chiều dữ liệu.

6. **Xây dựng mô hình**: Lựa chọn và huấn luyện mô hình.
7. **Đánh giá mô hình**: Kiểm tra hiệu suất và điều chỉnh.
8. **Triển khai và trình bày kết quả**: Áp dụng mô hình và truyền đạt kết quả.

Dưới đây, tôi sẽ áp dụng quy trình này để trả lời các yêu cầu của bạn, đặc biệt tập trung vào câu hỏi về xử lý ngoại lai IQR và hệ số tương quan Pearson, xử lý dữ liệu thiếu, cân chỉnh trung bình, và xây dựng mô hình dự báo.

Phần 1: Xác định việc xử lý ngoại lai với IQR có tương đồng với bước giảm biến nhiễu/dư thừa bằng hệ số tương quan Pearson

Bước 1: Xác định vấn đề

- **Mục tiêu**:
  - Dự báo biến gốc (giả định là `VNINDEX` hoặc một biến liên quan) dựa trên tập dữ liệu `VNINDEX_iqr_filled_HSG_Adjust.csv`.
  - Xác định xem xử lý ngoại lai với IQR có tương đồng với bước giảm biến nhiễu/dư thừa bằng hệ số tương quan Pearson trong pha giảm chiều cho PCA.
  - Xử lý dữ liệu thiếu, cân chỉnh trung bình, đảm bảo tính dừng, và chọn thành phần chính dựa trên phương sai tích lũy/giá trị riêng.
- **Ngữ cảnh**:
  - Dữ liệu: Chuỗi thời gian tài chính từ 2012-01-03 đến 2017-05-26, gồm 36 cột (`VNINDEX`, `BVH`, `CSM`, ..., `CONGNGHE`), đã xử lý ngoại lai với IQR (hệ số 3.0) và nội suy thiếu dữ liệu.
  - PCA được sử dụng để giảm chiều dữ liệu trước khi xây dựng mô hình dự báo.

Bước 2: Thu thập dữ liệu

- **Nguồn dữ liệu**: Tập `VNINDEX_iqr_filled_HSG_Adjust.csv`.
  - Đặc điểm: 36 cột số, index là ngày, cột `HSG` đã được thay bằng `HSG_log` (log1p).
  - Đã xử lý: Ngoại lai (IQR), dữ liệu thiếu (nội suy tuyến tính).
- **Vấn đề cần xác nhận**: So sánh xử lý ngoại lai (IQR) với bước giảm biến nhiễu/dư thừa (hệ số tương quan Pearson).

Bước 3: Khám phá dữ liệu (EDA)

- **Phân tích đặc điểm**:
  - **Kích thước**: ~1300 dòng (2012-01-03 đến 2017-05-26), 36 cột.
  - **Thang đo**:
    - `VNINDEX`: ~300-700.
    - `HSG_log`: ~0.9-3.4.

    - Các cột khác (`BVH` , `KDC` , ...): Thang đo khác nhau.
  - **Phân phối**: `HSG_log` đã giảm skewness nhờ log1p, nhưng các cột khác cần kiểm tra (ví dụ: bằng histogram).
  - **Tính dừng**: Chuỗi thời gian tài chính thường không dừng, cần kiểm tra (ADF test).
- **Vấn đề đã phát hiện**:
  - Ngày trùng lặp (2012-01-21 đến 2012-01-27 cho `HSG_log` = 1.098612).
  - Dữ liệu thiếu đã được nội suy, nhưng cần áp dụng trung bình trượt bậc 5 theo yêu cầu.
- **So sánh IQR và Pearson**:
  - **Xử lý ngoại lai với IQR**:
    - **Mục đích**: Loại bỏ hoặc thay thế các giá trị cực (ngoại lai) để làm mịn dữ liệu.
    - **Phương pháp**:
      - Tính $Q1, Q3, IQR = Q3 - Q1$.
      - Ngoại lai: $< Q1 - 3 \cdot IQR$ hoặc $> Q3 + 3 \cdot IQR$.
      - Thay thế: Giá trị biên hoặc log1p (cho `HSG_log`).
    - **Tác động**: Giảm ảnh hưởng của các điểm dữ liệu bất thường, ổn định trung bình và phương sai.
    - **Phạm vi**: Áp dụng trên từng cột riêng lẻ, không xem xét mối quan hệ giữa các cột.
  - **Giảm biến nhiễu/dư thừa bằng Pearson**:
    - **Mục đích**: Loại bỏ các biến có tương quan cao (dư thừa) hoặc không liên quan đến biến mục tiêu (nhiễu).
    - **Phương pháp**:
      - Tính hệ số tương quan Pearson giữa các cột:

$$\varrho_{jk} = \frac{\mathrm{Cov}(X_j, X_k)}{\sigma_j \sigma_k}$$

      - Loại biến:
        - Biến có $|\varrho_{jk}| \approx 1$: Dư thừa (chọn một trong hai).
        - Biến có $|\varrho_{j,\text{target}}| \approx 0$: Nhiễu (loại bỏ).
    - **Tác động**: Giảm số lượng biến, giữ lại các biến độc lập và có liên quan.
    - **Phạm vi**: Xem xét mối quan hệ giữa các cột và với biến mục tiêu.
  - **Kết luận về tương đồng**:
    - **Không tương đồng**:
      - **IQR**: Xử lý ngoại lai trong từng cột, không liên quan đến mối quan hệ giữa các cột.
      - **Pearson**: Giảm số lượng biến dựa trên tương quan giữa các cột hoặc với biến mục tiêu.

- **Mục đích khác nhau**:
  - IQR thuộc bước **làm sạch dữ liệu** (bước 4), nhằm cải thiện chất lượng dữ liệu.
  - Pearson thuộc bước **chuẩn bị dữ liệu** (bước 5), pha giảm chiều, nhằm giảm số biến trước PCA.
- **Thứ tự**:
  - IQR được thực hiện trước để đảm bảo dữ liệu sạch.
  - Pearson được thực hiện sau để chọn biến phù hợp cho PCA.

Bước 4: Làm sạch dữ liệu

- **Tình trạng thiếu dữ liệu**:
  - **Yêu cầu**: Sử dụng trung bình trượt bậc 5 (moving average, MA(5)) để nội suy dữ liệu thiếu.
  - **Phương pháp**:
    - Trung bình trượt bậc 5: Với mỗi giá trị thiếu tại t, lấy trung bình của 5 giá trị trước đó ($t-5$ đến $t-1$).
    - Công thức:

$$X_t = \frac{X_{t-5} + X_{t-4} + X_{t-3} + X_{t-2} + X_{t-1}}{5}$$

    - Nếu không đủ 5 giá trị trước (ví dụ: đầu chuỗi), sử dụng trung bình của các giá trị có sẵn hoặc nội suy tuyến tính.
  - **Mã Python**:

python Collapse Wrap Run Copy

```python
import pandas as pd

# Đọc dữ liệu gốc (giả định chưa nội suy)
df = pd.read_csv('VNINDEX_raw.csv', parse_dates=[0], index_col=0)

# Nội suy bằng trung bình trượt bậc 5
df_filled = df.copy()
for column in df.columns:
    # Thay NaN bằng trung bình trượt bậc 5
    df_filled[column] = df[column].fillna(
        df[column].rolling(window=5, min_periods=1).mean().shift(1)
    )
    # Nếu vẫn còn NaN (đầu chuỗi), nội suy tuyến tính
    df_filled[column] = df_filled[column].interpolate(method='linear')

# Lưu dữ liệu
df_filled.to_csv('VNINDEX_ma5_filled.csv')
```

- **Ngoại lai**: Đã xử lý bằng IQR (hệ số 3.0) trong `VNINDEX_iqr_filled_HSG_Adjust.csv`. Nếu cần xử lý lại:

```python
def iqr_outlier_treatment(series, factor=3.0):
    Q1 = series.quantile(0.25)
    Q3 = series.quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - factor * IQR
    upper_bound = Q3 + factor * IQR
    return series.clip(lower=lower_bound, upper=upper_bound)

df_cleaned = df_filled.apply(iqr_outlier_treatment)
```

- **Ngày trùng lặp**:
  - Kiểm tra và xử lý:

```python
print("Số ngày trùng lặp:", df_cleaned.index.duplicated().sum())
df_cleaned = df_cleaned[~df_cleaned.index.duplicated(keep='first')]
```

Bước 5: Chuẩn bị dữ liệu

- **Cân chỉnh trung bình (Mean Centering)**:
  - **Yêu cầu**: Đưa trung bình mỗi cột về 0 để tính thành phần chính.
  - **Phương pháp**:
    - Trừ trung bình μ_j của mỗi cột:

$$X'_{ij} = X_{ij} - \mu_j$$

  - **Z-score (khuyến nghị)**:
    - Vì các cột có thang đo khác nhau, áp dụng Z-score để chuẩn hóa:

$$Z_{ij} = \frac{X_{ij} - \mu_j}{\sigma_j}$$

    - Kết quả: $\mu_j = 0$, $\sigma_j = 1$, ma trận hiệp phương sai trở thành ma trận tương quan.
  - **Mã Python**:

```python
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
```

```
df_standardized = pd.DataFrame(scaler.fit_transform(df_cleaned),
                               columns=df_cleaned.columns,
                               index=df_cleaned.index)
df_standardized.to_csv('VNINDEX_standardized.csv')
```

- **Pha giảm chiều**:
  - **Bước 1: Giảm biến nhiễu/dư thừa bằng Pearson**:
    - Tính ma trận tương quan:

```
python                                   Collapse  Wrap  Run  Copy
correlation_matrix = df_standardized.corr()
```

    - Loại biến:
      - **Dư thừa**: Nếu $|\varrho_{jk}| > 0.8$ (ngưỡng tùy chọn), giữ một trong hai biến.
      - **Nhiễu**: Nếu $|\varrho_{j,\text{VNINDEX}}| < 0.1$ (giả định `VNINDEX` là mục tiêu), loại biến.
    - **Mã Python**:

```
python                                   Collapse  Wrap  Run  Copy
import numpy as np

# Loại biến dư thừa
threshold = 0.8
upper = correlation_matrix.where(np.triu(np.ones(correlation_matrix.shape), k=1
to_drop = [col for col in upper.columns if any(upper[col].abs() > threshold)]

# Loại biến nhiễu (tương quan thấp với VNINDEX)
target_corr = correlation_matrix['VNINDEX'].abs()
to_drop += [col for col in target_corr.index if target_corr[col] < 0.1 and col

# Giữ các biến không dư thừa/nhiễu
df_reduced = df_standardized.drop(columns=to_drop)
df_reduced.to_csv('VNINDEX_reduced.csv')
```

  - **Bước 2: PCA**:
    - Áp dụng PCA trên `df_reduced`.
    - **Chọn số thành phần chính**:
      - Dựa trên **tỷ lệ phương sai tích lũy** (ví dụ: giữ 95% phương sai).
      - Kết hợp **giá trị riêng** (eigenvalues > 1, theo tiêu chuẩn Kaiser).
    - **Mã Python**:

```
python                                   Collapse  Wrap  Run  Copy
```

```python
from sklearn.decomposition import PCA

pca = PCA()
pca.fit(df_reduced)

# Tỷ lệ phương sai tích lũy
cumulative_variance = np.cumsum(pca.explained_variance_ratio_)
n_components = np.argmax(cumulative_variance >= 0.95) + 1

# Kiểm tra giá trị riêng
eigenvalues = pca.explained_variance_
n_components = min(n_components, sum(eigenvalues > 1))

print(f"Số thành phần chính: {n_components}")

# Biến đổi dữ liệu
pca = PCA(n_components=n_components)
df_pca = pd.DataFrame(pca.fit_transform(df_reduced),
                      index=df_reduced.index,
                      columns=[f'PC{i+1}' for i in range(n_components)])
df_pca.to_csv('VNINDEX_pca.csv')
```

- **Xử lý tính dừng**:
  - **Yêu cầu**: Đảm bảo các biến dừng trước khi xây dựng mô hình dự báo.
  - **Phương pháp**:
    - Kiểm tra tính dừng bằng **Augmented Dickey-Fuller (ADF) test**:
      - H_0: Chuỗi không dừng.
      - Nếu p-value > 0.05, chuỗi không dừng.
    - Nếu không dừng, lấy **sai phân bậc 1**:

$$\Delta X_t = X_t - X_{t-1}$$

    - Lặp lại ADF test trên chuỗi sai phân.
  - **Mã Python**:

python Collapse Wrap Run Copy

```python
from statsmodels.tsa.stattools import adfuller

def check_stationarity(series, name):
    result = adfuller(series.dropna())
    print(f"{name} ADF p-value: {result[1]}")
    return result[1] <= 0.05

# Kiểm tra tính dừng
for column in df_reduced.columns:
```

```
    if not check_stationarity(df_reduced[column], column):
        # Lấy sai phân bậc 1
        df_reduced[column] = df_reduced[column].diff().fillna(0)
        check_stationarity(df_reduced[column], f"{column}_diff")

# Lưu dữ liệu dừng
df_reduced.to_csv('VNINDEX_stationary.csv')
```

Bước 6: Xây dựng mô hình

- **Pha xây dựng mô hình và dự báo**:
  - **Mô hình**: Sử dụng mô hình chuỗi thời gian (ví dụ: ARIMA, VAR) hoặc hồi quy (ví dụ: Linear Regression, LSTM) trên dữ liệu PCA (`df_pca`) để dự báo `VNINDEX` .
  - **Ví dụ với ARIMA**:

python Collapse Wrap Run Copy

```
from statsmodels.tsa.arima.model import ARIMA

# Dự báo VNINDEX (giả định PC1 liên quan mạnh)
model = ARIMA(df_pca['PC1'], order=(1,1,1))
model_fit = model.fit()
forecast = model_fit.forecast(steps=10)
print("Dự báo 10 ngày:", forecast)
```

Bước 7: Đánh giá mô hình

- **Đo lường**:
  - Sử dụng RMSE, MAE để đánh giá sai số dự báo.
  - So sánh với dữ liệu thực (nếu có).
- **Mã Python**:

python Collapse Wrap Run Copy

```
from sklearn.metrics import mean_squared_error

# Giả định có dữ liệu thực
actual = df_pca['PC1'][-10:]
rmse = np.sqrt(mean_squared_error(actual, forecast))
print("RMSE:", rmse)
```

Bước 8: Triển khai và trình bày kết quả

- **Kết quả**:
  - Dữ liệu chuẩn hóa: `VNINDEX_standardized.csv` .
  - Dữ liệu giảm chiều: `VNINDEX_reduced.csv` , `VNINDEX_pca.csv` .

- Dự báo: Chuỗi dự báo cho `VNINDEX` hoặc thành phần chính.

- **Trình bày**:
  - Biểu đồ: Dữ liệu gốc, dữ liệu PCA, và dự báo.
  - Báo cáo: Tỷ lệ phương sai tích lũy, số thành phần chính, RMSE.

Phần 2: Phương pháp xử lý và khắc phục

2.1. Xử lý dữ liệu thiếu bằng trung bình trượt bậc 5

- **Phương pháp**: Đã trình bày ở bước 4 (làm sạch dữ liệu).
- **Khắc phục**:
  - Đảm bảo không còn NaN sau nội suy:

```python
print("Số giá trị thiếu:", df_filled.isna().sum().sum())
```

  - Nếu vẫn còn NaN, sử dụng nội suy tuyến tính hoặc giá trị trung bình cột.

2.2. Cân chỉnh trung bình tập dữ liệu

- **Phương pháp**: Đã trình bày ở bước 5 (Z-score bao gồm Mean Centering).
- **Khắc phục**:
  - Kiểm tra trung bình sau Z-score:

```python
print("Trung bình sau Z-score:", df_standardized.mean())
```

  - Đảm bảo trung bình ~ 0 (sai số làm tròn ~ 10^{-15}).

Phần 3: Xây dựng mô hình và dự báo (2 pha)

3.1. Pha giảm chiều

- **Bước 1: Giảm biến nhiễu/dư thừa bằng Pearson**: Đã trình bày ở bước 5.
- **Bước 2: PCA**: Đã trình bày ở bước 5, với lựa chọn thành phần chính dựa trên:
  - Tỷ lệ phương sai tích lũy ≥ 95%.
  - Giá trị riêng > 1.

3.2. Pha xây dựng mô hình và dự báo

- **Mô hình**: ARIMA trên thành phần chính (PC1, PC2, ...).
- **Dự báo**: Dự báo 10 ngày tiếp theo.
- **Mã Python**: Đã cung cấp ở bước 6.

3.3. Xử lý tính dừng

- **Phương pháp**: ADF test và sai phân, đã trình bày ở bước 5.
- **Khắc phục**:
  - Nếu chuỗi vẫn không dừng sau sai phân bậc 1, thử sai phân bậc 2 hoặc chuyển đổi log.

Phần 4: Chọn số lượng thành phần chính

- **Phương pháp**:
  - **Tỷ lệ phương sai tích lũy**: Chọn số thành phần sao cho $\sum \text{variance ratio} \geq 0.95$.
  - **Giá trị riêng**: Giữ thành phần có $\lambda_i > 1$.
- **Mã Python**: Đã cung cấp ở bước 5.
- **Ví dụ kết quả**:
  - Nếu `cumulative_variance = [0.6, 0.85, 0.96, ...]`, chọn 3 thành phần.
  - Nếu `eigenvalues = [5.2, 2.1, 0.9, ...]`, chọn 2 thành phần ($\lambda > 1$).
  - Lấy **tối thiểu** của hai tiêu chí.

Kết luận

1. **IQR vs Pearson**:
   - Không tương đồng. IQR xử lý ngoại lai (làm sạch dữ liệu), Pearson giảm biến nhiễu/dư thừa (giảm chiều).
   - IQR thuộc bước 4, Pearson thuộc bước 5.
2. **Xử lý dữ liệu thiếu**: Sử dụng trung bình trượt bậc 5, kết hợp nội suy tuyến tính nếu cần.
3. **Cân chỉnh trung bình**: Áp dụng Z-score để chuẩn hóa ($\mu_j = 0$, $\sigma_j = 1$).
4. **Mô hình dự báo**:
   - Pha 1: Giảm chiều bằng Pearson và PCA.
   - Pha 2: Xây dựng ARIMA trên dữ liệu PCA, đảm bảo tính dừng.
5. **Chọn thành phần chính**: Kết hợp ngưỡng phương sai tích lũy (95%) và giá trị riêng (>1).

Nếu bạn cần mã chi tiết hơn, kiểm tra dữ liệu, hoặc triển khai cụ thể, hãy cho tôi biết!