INPUT:

params:

- n\_segments: Số đoạn phân tích tối đa

- vertex\_count\_overshoot: Số điểm breakpoints vượt quá cho phép

- recovery\_threshold: Ngưỡng phục hồi

- minObservation: Số lượng quan sát tối thiểu

- human\_affected: Boolean, xác định nếu khu vực bị con người tác động

data:

- NDVI: Dữ liệu NDVI theo thời gian

- pixel\_id: ID của từng pixel

OUTPUT:

- File Excel chứa thông tin về các phân đoạn của tất cả các pixel.

- Các hình ảnh minh họa quá trình phân đoạn của từng pixel.

QUÁ TRÌNH:

MAIN FUNCTION:

1. Load dữ liệu NDVI từ tệp đầu vào.(all\_data)

2. Khởi tạo tham số `params`:

- n\_segments = 7

- vertex\_count\_overshoot = 3

- recovery\_threshold = 0.25

- minObservation = 7

- human\_affected = True

3. Chạy xử lý song song trên các pixel bằng `run\_parallel\_with\_client`.

---

FUNCTION run\_parallel\_with\_client(params, all\_data):

1. Với mỗi pixel trong dữ liệu all\_data:

a. Gửi pixel\_id và dữ liệu NDVI vào hàm `process\_pixel\_dask`.

b. Tạo một danh sách các task [`process\_pixel\_dask`] ở trạng thái delayed.

2. Phân chia tác vụ ‘process\_pixel\_dask’ song song với Dask.

3. Kết thúc khi tất cả các pixel được xử lý.

---

FUNCTION process\_pixel\_dask(pixel\_id, params, data):

1. \*\*Kiểm tra dữ liệu:\*\*

- Nếu dữ liệu có số quan sát < `minObservation`, bỏ qua pixel.

2. \*\*Xử lý dữ liệu:\*\*

- Khởi tạo một đối tượng detector lớp `ChangeDetection` với params, pixel\_id, và NDVI của pixel.

- detector.smooth\_spikes()

3. \*\*Phân đoạn dữ liệu:\*\*

- detector.segment\_with\_overshoot()

- detector. filter\_breakpoints\_dynamic()

- detector. remove\_breakpoint\_with\_max\_variance

4. \*\*Xử lý phục hồi:\*\*

- Nếu `human\_affected` = False: => tức quá trình tự nhiên không có sự tác động của con người.

a. detector. detect\_and\_remove\_short\_term\_recovery();

b. detector. detect\_and\_remove\_small\_recoveries();

5. \*\*Xuất kết quả:\*\*

- detector.get\_segment\_data()

- detector.plot\_results()

---

CLASS ChangeDetection:

\*\*Constructor:\*\*

- Nhận params, pixel\_id, và chuỗi NDVI.

- Khởi tạo các biến cần thiết: breakpoints, slopes, pwlf\_model.

1. FUNCTION smooth\_spikes(self):

Sử dụng Savitzky-Golay Filter làm mịn chuỗi dữ liệu.

2. FUNCTION **segment\_with\_overshoot**(self):

Hồi quy phân đoạn dữ liệu

3. FUNCTION filter\_breakpoints\_dynamic(breakpoints, slopes, ndvi\_values):

a. Tính toán ngưỡng động dựa trên độ thay đổi NDVI.

b. Lọc các điểm ngắt không hợp lệ:

- Nếu không đạt ngưỡng động, loại bỏ điểm ngắt.

c. Trả về danh sách điểm ngắt đã lọc.

4. FUNCTION remove\_breakpoint\_with\_max\_variance(time\_numeric, ndvi\_values, max\_segments):

a. Trong khi số breakpoints > max\_segments:

- Tìm điểm ngắt làm tăng phương sai lớn nhất.

- Loại bỏ điểm ngắt này.

b. Cập nhật danh sách điểm ngắt và mô hình.

5. FUNCTION detect\_and\_remove\_short\_term\_recovery():

a. Duyệt qua các đoạn phân đoạn.

b. Loại bỏ phục hồi ngắn hạn:

- Phục hồi xảy ra trong thời gian rất ngắn (1 năm).

- Phục hồi có tốc độ nằm trong ngưỡng nhỏ.

6. FUNCTION detect\_and\_remove\_small\_recoveries():

a. Duyệt qua các đoạn phân đoạn.

b. Loại bỏ các phục hồi không đáng kể:

- Phục hồi có độ lớn thấp hơn `recovery\_threshold`.

7. FUNCTION plot\_result(pixel\_id):

a. Vẽ đồ thị dữ liệu gốc, dữ liệu làm mịn, và các đoạn hồi quy.

b. Lưu biểu đồ với tên chứa pixel\_id.

8. FUNCTION get\_segment\_data(pixel\_id):

a. Tính toán thông tin từng đoạn:

- Thời gian, giá trị bắt đầu/kết thúc, tốc độ thay đổi.

b. Xuất thông tin vào Excel.