



TS_NDVI

Created by	Luciana Nieto
Tags	AgriData

- Filtra una colección de imágenes satelitales Sentinel-2 por fecha y ubicación geográfica, utilizando un polígono definido.
- Aplica un filtro de nubes a las imágenes, eliminando los píxeles nublados.
- Calcula el Índice de Vegetación de Diferencia Normalizada (NDVI) para cada imagen.
- Genera puntos aleatorios dentro del polígono y muestra una serie temporal del NDVI para cada punto.
- Muestra el gráfico de la serie temporal y agrega los puntos al mapa.

En resumen, esta parte del código permite analizar un solo campo (representado por el polígono) y examinar la variación del NDVI a lo largo del tiempo en diferentes ubicaciones dentro del campo.

Parte 2:

- Filtra una colección de imágenes Sentinel-2 por fecha y ubicación geográfica, utilizando una colección de múltiples polígonos.
- Aplica un filtro de nubes y calcula el NDVI para cada imagen, de manera similar a la Parte 1.
- Genera puntos aleatorios para cada polígono y suaviza la serie temporal del NDVI utilizando un filtro de promedio móvil.
- Muestra el gráfico suavizado de la serie temporal.

En esta parte, el código permite realizar un análisis similar al de la Parte 1, pero en lugar de un solo campo, se analizan varios campos representados por los polígonos en la colección. La suavización de la serie temporal proporciona una visión más general de las tendencias del NDVI en cada campo a lo largo del tiempo.

En general, este código muestra cómo utilizar GEE para filtrar imágenes satelitales, calcular índices de vegetación, generar puntos aleatorios y realizar análisis de series temporales en campos agrícolas o áreas de interés.

```
//para un solo campo varios puntos random

var poligono = geometry

// Filtrar la colección de imágenes Sentinel-2
var colección = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
  .filterDate('2017-01-01', '2019-12-31') // Ajusta el rango de fechas
  .filterBounds(poligono);

// Función para aplicar el filtro de nubes
var maskClouds = function(image) {
  var cloudMask = image.select('QA60').not();
  return image.updateMask(cloudMask);
};
```

```

// Función para añadir NDVI
function addNDVI(image) {
  var ndvi = image.normalizedDifference(['B8', 'B4']).rename('ndvi');
  return image.addBands(ndvi);
}

// random points en poligono
var puntos = ee.FeatureCollection.randomPoints(poligono, 6);

// Aplica las funciones de filtrado de nubes y NDVI a las imágenes
var colecciónFiltrada = colección.map(maskClouds).map(addNDVI);

// Extrae el valor NDVI para la serie temporal sobre cada uno de los puntos
var chart = ui.Chart.image.seriesByRegion({
  imageCollection: colecciónFiltrada.select('ndvi'),
  regions: puntos,
  reducer: ee.Reducer.mean(),
  scale: 10,
  xProperty: 'system:time_start'
});

print(chart);

// Añadir los puntos al mapa
Map.addLayer(puntos, {}, 'Random Points0');

// Definir múltiples polígonos
//var poligonos = ee.FeatureCollection([geometry, geometry2, geometry3, geometry4]);
var poligonos = ee.FeatureCollection([geometry5]);

// Filtrar la colección de imágenes Sentinel-2
var colección = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
  .filterDate('2018-01-01', '2022-12-31') // Ajusta el rango de fechas
  .filterBounds(poligonos);

// filtro de nubes
var maskClouds = function(image) {
  var cloudMask = image.select('QA60').not();
  return image.updateMask(cloudMask);
};

// Función para añadir NDVI
function addNDVI(image) {
  var ndvi = image.normalizedDifference(['B8', 'B4']).rename('ndvi');
  return image.addBands(ndvi);
}

// Generar una colección de puntos aleatorios para cada polígono
var points = poligonos.map(function(poly) {
  var randomPoints = ee.FeatureCollection.randomPoints({
    region: poly.geometry(),
    points: 1
  });
  return randomPoints;
}).flatten();

// Aplica las funciones de filtrado de nubes, NDVI
var colecciónFiltrada = colección.map(maskClouds).map(addNDVI);

// Reductor para aplicar un filtro de promedio móvil
var reducer = ee.Reducer.mean();

// Aplicar un filtro de promedio móvil a la serie temporal
var smoothedChart = ui.Chart.image.seriesByRegion({
  imageCollection: colecciónFiltrada.select(['ndvi']),
  regions: poligonos,
  reducer: reducer,
  scale: 10,
  xProperty: 'system:time_start'
});

```

```

}).setOptions({title: 'Serie Temporal Suavizada'});

// Muestra las gráficas
print('Gráfico suavizado:', smoothedChart);

////////////////////////////////////

var startDate = ee.Date('2018-01-01');
var endDate = ee.Date('2022-12-30');
var interval = 6; // Definir el intervalo en meses

// Crear una secuencia de números que representan los semestres desde 2018 hasta 2023
var n = endDate.difference(startDate, 'month').divide(interval).ceil();
var sequence = ee.List.sequence(0, n.subtract(1));

// Crear una colección de imágenes que contienen los valores mínimos y máximos de NDVI y EVI para cada semestre y punto aleatorio
var imageCollection = ee.ImageCollection(sequence.map(function(i) {
  // Crear el rango de fechas para este semestre
  var start = startDate.advance(ee.Number(i).multiply(interval), 'month');
  var end = start.advance(interval, 'month');

  // Filtrar la colección de imágenes por este semestre
  var filtered = colecciónFiltrada.filterDate(start, end);

  // Reducir la colección de imágenes a una sola imagen que contenga los valores mínimos y máximos
  var minMax = filtered.reduce(ee.Reducer.minMax());

  // Añadir las fechas como bandas a la imagen
  minMax = minMax.addBands(ee.Image.constant(start.millis()).rename('start'));
  minMax = minMax.addBands(ee.Image.constant(end.millis()).rename('end'));

  return minMax;
})));

// Convertir la colección de imágenes en una colección de features
var featureCollection2 = imageCollection.map(function(image) {
  return image.reduceRegions({
    collection: poligonos,
    reducer: ee.Reducer.firstNonNull(),
    scale: 10
  }).map(function(feature) {
    // Convertir las fechas de milisegundos a un formato de fecha legible
    var start = ee.Date(feature.get('start')).format('YYYY-MM-dd');
    var end = ee.Date(feature.get('end')).format('YYYY-MM-dd');

    // Asignar los valores de NDVI y EVI a las propiedades de las features
    return feature
      .set('start', start)
      .set('end', end)
      .set('ndvi_min', feature.get('ndvi_min'))
      .set('ndvi_max', feature.get('ndvi_max'));
  });
}).flatten();

print('Valores máximos y mínimos:', featureCollection2);

// Añadir los puntos al mapa
Map.addLayer(points, {}, 'Random Points2');

var start = '2017-01-01';
var end = '2022-12-31';
var collection = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
  .filterBounds(geometry4)
  .filterDate(start, end);

var dates = ee.List(collection.aggregate_array('system:time_start')).map(function(time) {
  return ee.Date(time).format('YYYY-MM-dd');
});

```

```

}).getInfo();

print('Dates: ', dates);

////viz

var image = ee.ImageCollection('COPERNICUS/S2_SR')
  .filterBounds(geometry5)
  .filterDate('2019-11-02', '2019-11-08')
  .sort('CLOUD_COVER', false);

var conteo = image.size();
print('conteo de imágenes', conteo);

var mejorImagen = ee.Image(image.sort('CLOUD_COVER').first());

print('La primera con menos nubosidad', mejorImagen);

Map.addLayer(mejorImagen.clip(geometry5), {bands: ['B4', 'B3', 'B2'], max: 3000}, 'image');

```

