

In []:

```
// Select the specific field from the allRegions (asset containing the 4 fields received from Manna)
var region = ee.FeatureCollection(allRegions.filterMetadata("system:index", "equals", "00000000000000000003"));
// Example of selecting field number 4

// Outputs the FeatureCollection of the 5-days batch to a CSV file on Google Drive.
function one_Batch(startDate, numberOfDays, index){
  var dayOffsets = ee.List.sequence(0, numberOfDays - 1)
  var allDaily = ee.ImageCollection(dayOffsets.map(function (dayOffset) {
    var start = startDate.advance(dayOffset, 'days')
    var end = start.advance(1, 'days')
    var filter = ee.Filter.and(
      ee.Filter.date(start, end),
      ee.Filter.bounds(region)
    )
    var s2 = ee.ImageCollection('COPERNICUS/S2_SR')
      .filter(filter)
      .map(function (image) {
        return image
          .normalizedDifference(['B8', 'B4'])
          .updateMask(image.select('QA60').not())
          .rename('S2')
      })
      .median()

    var s3 = ee.ImageCollection('COPERNICUS/S3/OLCI')
      .filter(filter)
      .map(function (image) {
        return image
          .select(['Oa17_radiance', 'Oa08_radiance'])
          .divide(ee.Image([0.00493004, 0.00876539]))
          .normalizedDifference(['Oa17_radiance', 'Oa08_radiance'])
          .rename('S3')
      })
      .median()
    return s2
      .addBands(s3)
      .set('date', start.format('yyyy-MM-dd'))
  }))

  var completeDaily = allDaily
    .sort('date')
    .toList(numberOfDays)

  print('Make sure there are ' + numberOfDays + ' images:', completeDaily.size())

  var firstS2Image = completeDaily.get(0)
  var lastS2Image = completeDaily.get(numberOfDays - 1)

  // First Sentinel2 image NDVI values (T0)
  var firstS2Feature = toFeature('No', 'S2', 'T0', firstS2Image)

  // Validation Sentinel2 image NDVI values (T5)
  var lastS2Feature = toFeature('Validation', 'S2', 'T' + (numberOfDays - 1), lastS2Image)

  // All the Sentinel3 images NDVI values zipped together.
```

```
var allS3Features = completeDaily
  .zip(ee.List.sequence(0, completeDaily.size().subtract(1)))
  .map(function (tuple) {
    tuple = ee.List(tuple)
    var time = ee.String('T').cat(tuple.getNumber(1).format('%d'))
    var image = tuple.get(0)
    return toFeature('No', 'S3', time, image)
  })

var collection = ee.FeatureCollection(fistS2Feature)
  .merge(allS3Features)
  .merge(ee.FeatureCollection(lastS2Feature))

function toFeature(validation, sensor, time, ndvi) {
  ndvi = ee.Image(ndvi)
  var pixels = ee.Number(ndvi
    .select(sensor)
    .reduceRegion({
      reducer: ee.Reducer.toList(),
      geometry: region,
      scale: 10
    })
    .values()
    .get(0))
  return ee.Feature(null, {
    VALIDATION: validation,
    SENSOR: sensor,
    TIME: time,
    NDVI: pixels
  })
}

// Outputs T0-T5 to drive.
Export.table.toDrive({
  collection: collection,
  description: ''+i,
  folder: 'msc_data/field4',
  selectors: ['VALIDATION', 'SENSOR', 'TIME', 'NDVI']
})
}
```

In []:

```

var start = ee.Date("2018-04-01")
var finish = ee.Date("2020-04-30")

///----- SENTINEL2 Collection -----
-///
var sen2 = ee.ImageCollection('COPERNICUS/S2_SR').filterDate(start, finish).filterBound
s(region)
  .map(function(image){
    return image
    .clipToCollection(region)
    .updateMask(image.select('QA60').not())
  })

// get all system:time_start in a list
var listMillis = ee.List(sen2.aggregate_array('system:time_start'));
var uniqueDates = listMillis.distinct();

sen2 = sen2.map(function(image){
  return image.set('simpleDateMillis', ee.Date(image.date().format('YYYY-MM-dd')).milli
s())
});
var listMillisSimple = ee.List(sen2.aggregate_array('simpleDateMillis'));

// Using only distinct dates, no repeats
var uniqueDatesSimple = listMillisSimple.distinct();

// Creating an ImageCollection that function one_Batch will iterate over
var SEN2_day0 = uniqueDatesSimple.map(function(date){
  return ee.Date(date).format();
});

for(var i=0 ; i<250 ; i++){
  var coll = one_Batch(ee.Date(SEN2_day0.get(i)), 6, i)
}

```