Part 2: Mangrove Extent Mapping and Time Series

Abigail Barenblitt & Temilola Fatoyinbo

Nov 12th, 2020

Course Structure and Materials

- Three, 1.5-hour sessions on November 5, 12, and 19
- The same content will be presented at two different times each day:
  - Session A: 10:00-11:30 EST (UTC-5)
  - Session B: 15:00-16:30 EST (UTC-5)

Course Structure and Materials

- Webinar recordings, PowerPoint presentations, and the homework assignment can be found after each session at:
  - https://appliedsciences.nasa.gov/join-mission/training/english/remote-sensing-mangroves-support-un-sustainable-development-goals
  - Q&A following each lecture and/or by email at:
    - lola.fatoyinbo@nasa.gov or
    - abigail.barenblitt@nasa.gov

Homework and Certificates

- **Homework:**
  - Three homework assignments, assigned after each weekly Part
  - Answers must be submitted via Google Forms

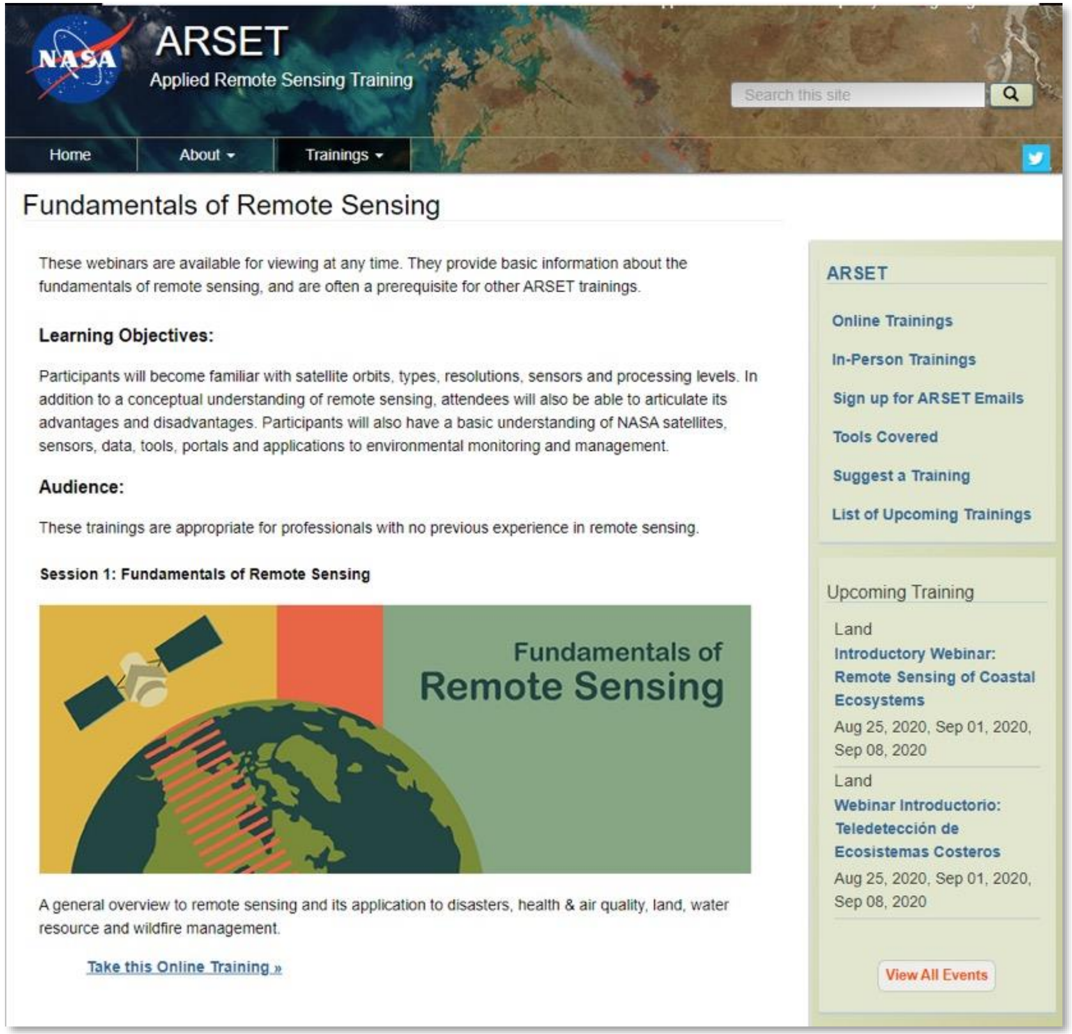
- **Certificate of Completion:**
  - Attend all three live webinars
  - Complete the homework assignments by the deadline (access from ARSET website)
  - You will receive certificates approximately two months after the completion of the course from: marines.martins@ssaihq.com

Prerequisites

- Required Version of QGIS: 3.10 https://www.qgis.org/en/site/forusers/download.html
- Download and Install Class Accuracy Plug-in for QGIS: https://github.com/remotesensinginfo/classaccuracy
- For instructions for installation refer to this video: https://www.youtube.com/watch?v=NJRdKpmujRo
- Fundamentals of Remote Sensing
- Intro to JavaScript for GEE
- Create a Google Earth Engine Account

Optional:

- GEE Beginner's Cookbook
- GEE Managing Assets
- Introduction to Google Earth Engine Tutorial



Learning Objectives

By the end of this presentation, you will:

- Understand the basics of using Google Earth Engine
- Create a mangrove extent map using a Random Forest Classification
- Create a time series for mangrove extent change

Outline

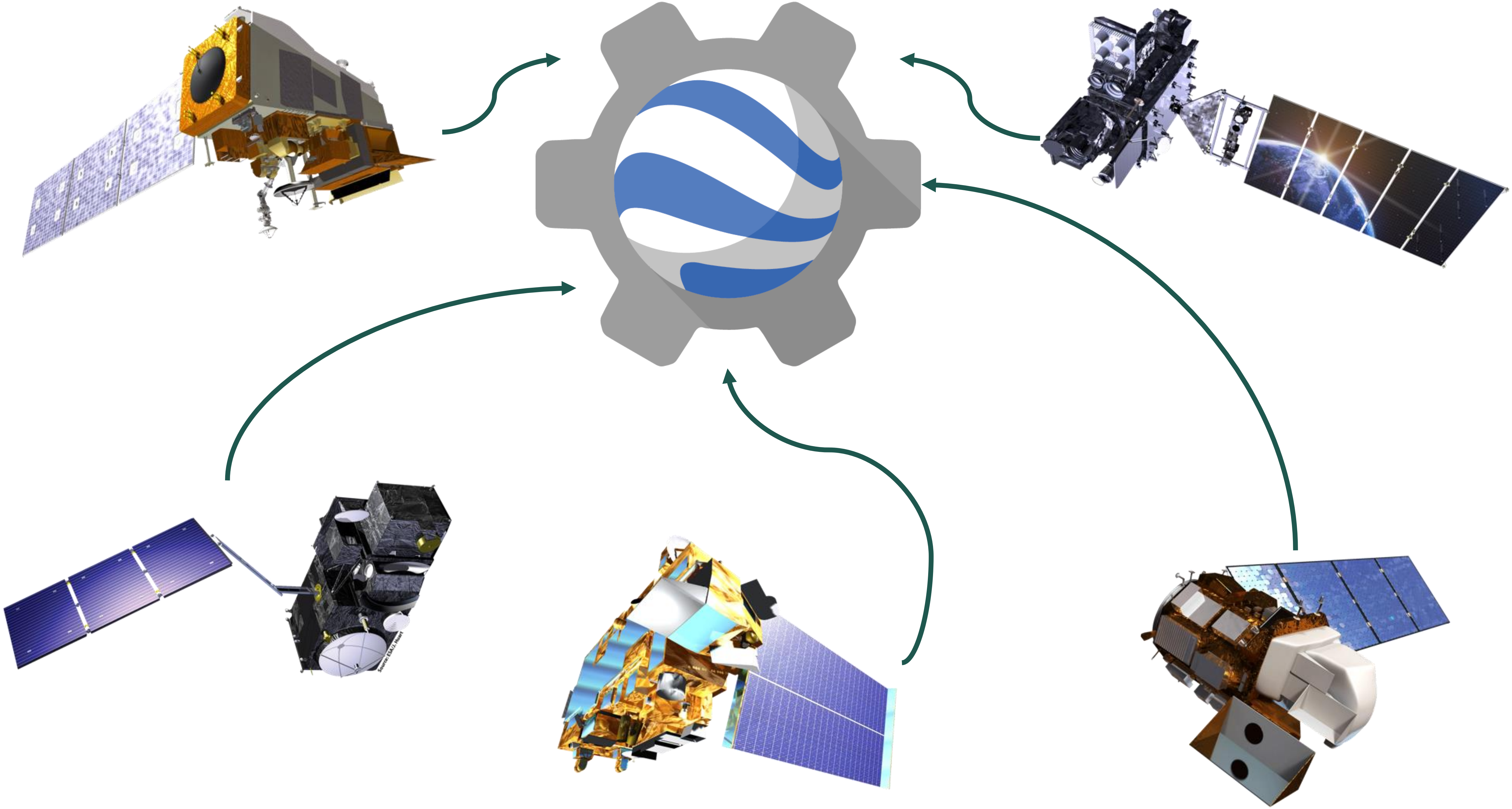
1) Review of Google Earth Engine
2) Review of Time Series Analysis
3) Demo of Time Series Analysis for Guyana
   – Setting Up the Map
   – Filtering a Landsat Composite
   – Constructing a Random Forest Model
   – Time Series Comparison
   – New Random Forest
   – Calculating Mangrove Area
   – Exporting Layers of Interest
   – Demo of QGIS Class Accuracy Plug-in

What is Google Earth Engine?

- A free, open-source, “cloud-based geospatial processing platform”
- Comprised of:
  - A catalog of publicly available datasets
  - Google’s computation power
  - An Application Programming Interface (API)
  - A code editor

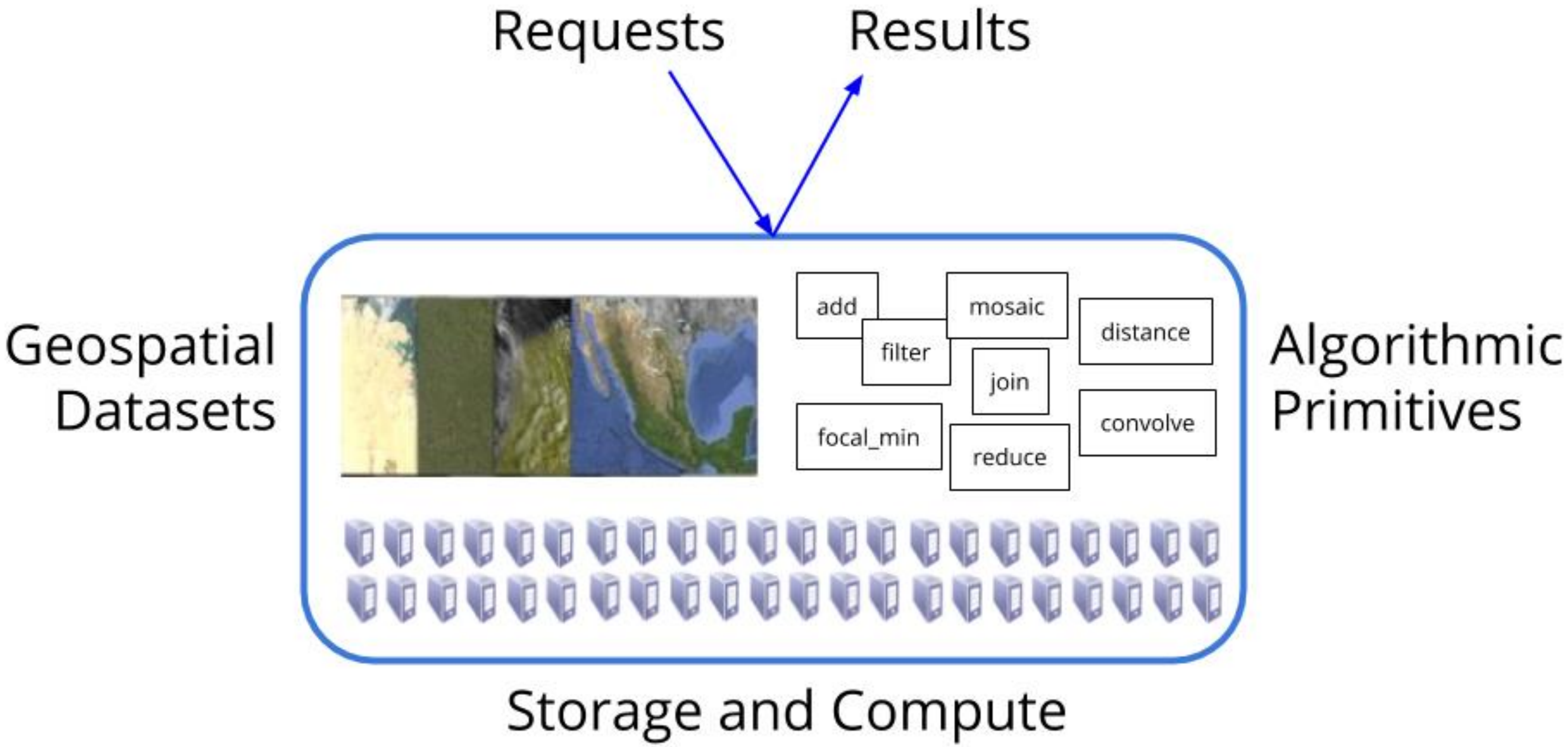
Google’s Mission:

Our mission is to organize the world’s information and make it universally accessible and useful.



How GEE Works

- Cloud-Based
  - Client vs. server functions
  - Users manipulate “proxy” objects through the server
  - Instructions sent to Google for processing and results sent back to web browser for display
- Defaults to WGS84 Projections
- Capabilities/Limitations
  - Planetary-scale
  - Quota restrictions due to open-source nature
  - User memory limited



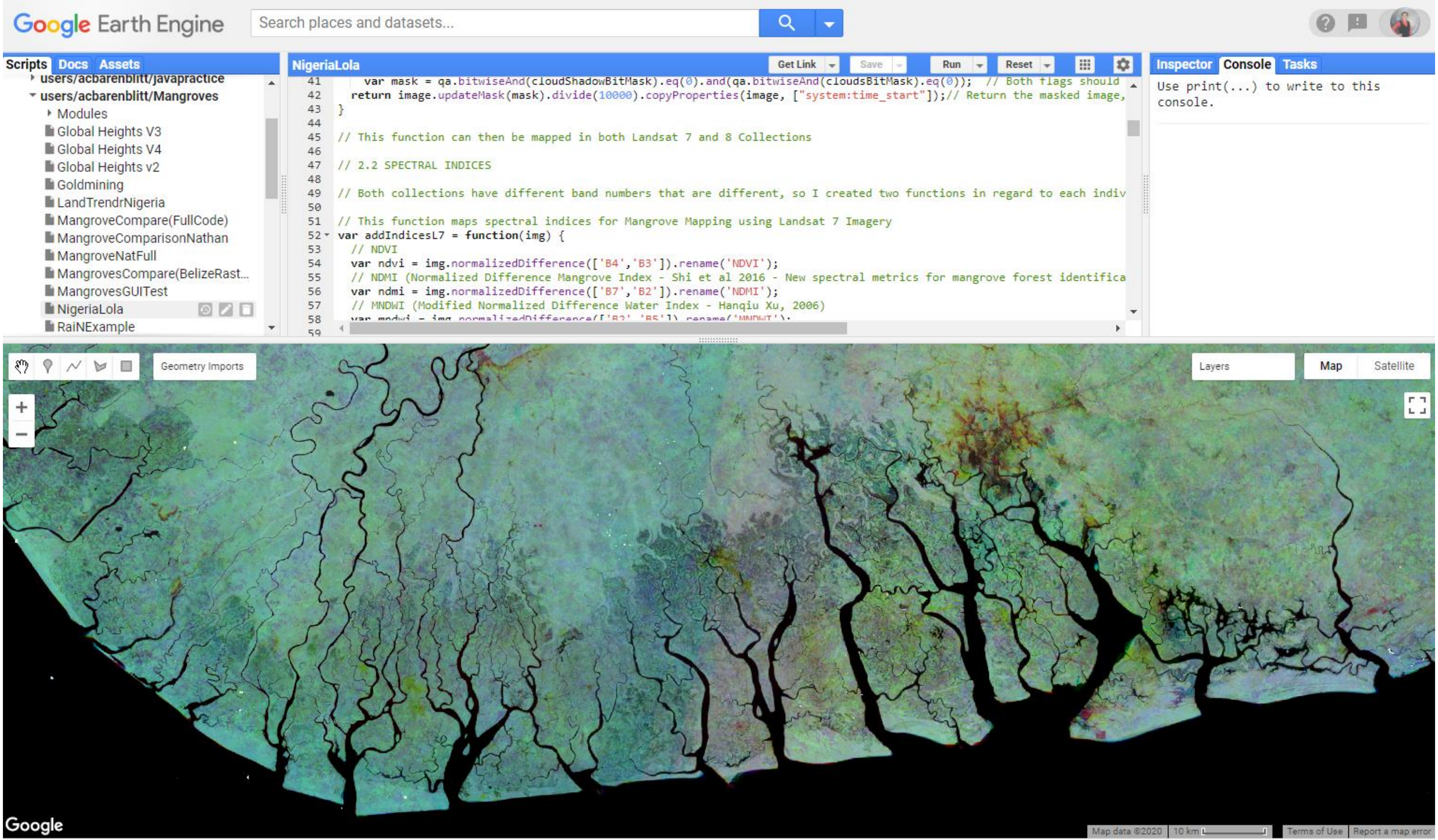
Google Earth Engine

The Code Editor

- For more detailed analyses
- JavaScript code editor (Python available)
- Map display
- API reference documentation (Docs tab)
- Console output (Console tab)
- Task Manager (Tasks tab)
- Interactive map query (Inspector tab)

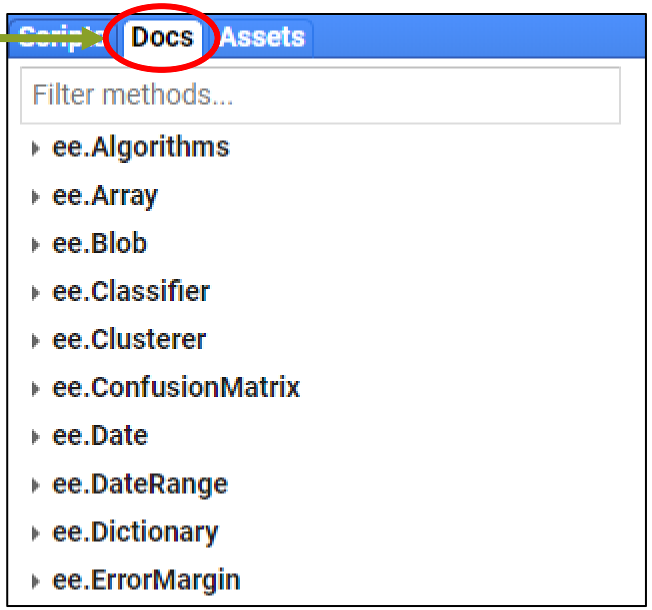
Glossary of Terms

- **Google Earth Engine Asset**
  - External dataset loaded into Google Earth Engine for analysis
- **Table**
  - Vector data in shapefile format
  - Example: Ground-truthed location data
- **Image**
  - Raster data composed of one or more bands
  - Example: Euclidean distance to stream
- **Image Collection**
  - A stack or time series of images
  - Example: Landsat 8 imagery


Google Earth Engine
Search places and datasets...
Scripts Docs Assets
users/acbarenblitt/javapractice
users/acbarenblitt/Mangroves
Modules
Global Heights V3
Global Heights V4
Global Heights v2
Goldmining
LandTrendrNigeria
MangroveCompare(FullCode)
MangroveComparisonNathan
MangroveNatFull
MangrovesCompare(BelizeRast...
MangrovesGUITest
NigeriaLola
RaiNExample
NigeriaLola
Get Link
Save
Run
Reset
var mask = qa.bitwiseAnd(cloudShadowBitMask).eq(0).and(qa.bitwiseAnd(cloudsBitMask).eq(0)); // Both flags should
return image.updateMask(mask).divide(10000).copyProperties(image, ["system:time_start"]);// Return the masked image,
}
// This function can then be mapped in both Landsat 7 and 8 Collections
// 2.2 SPECTRAL INDICES
// Both collections have different band numbers that are different, so I created two functions in regard to each indiv
// This function maps spectral indices for Mangrove Mapping using Landsat 7 Imagery
var addIndicesL7 = function(img) {
// NDVI
var ndvi = img.normalizedDifference(['B4','B3']).rename('NDVI');
// NDMI (Normalized Difference Mangrove Index - Shi et al 2016 - New spectral metrics for mangrove forest identifica
var ndmi = img.normalizedDifference(['B7','B2']).rename('NDMI');
// MNDWI (Modified Normalized Difference Water Index - Hanqiu Xu, 2006)
Inspector Console Tasks
Use print(...) to write to this console.
Geometry Imports
Layers
Map
Satellite
Google
Map data ©2020
10 km
Terms of Use
Report a map error

Resources for Help

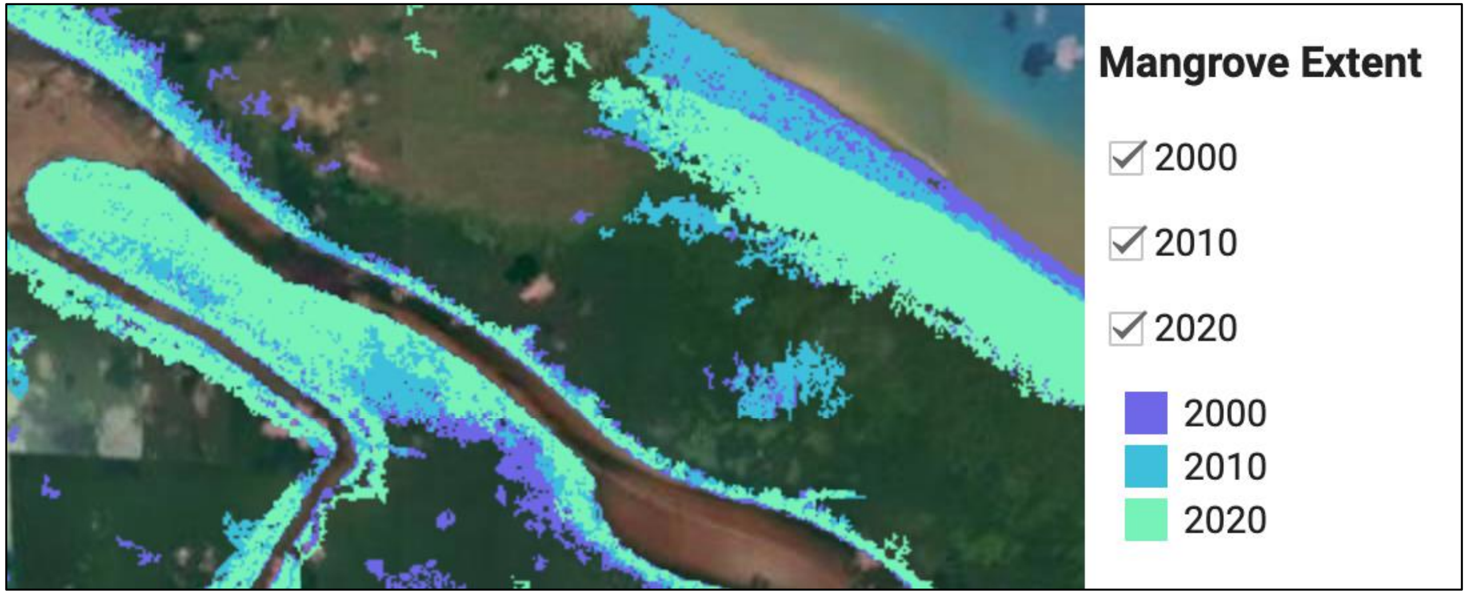
- Docs Tab
- Developer's Guide
- Google Earth Engine Developers Group



Time Series of Mangrove Extent

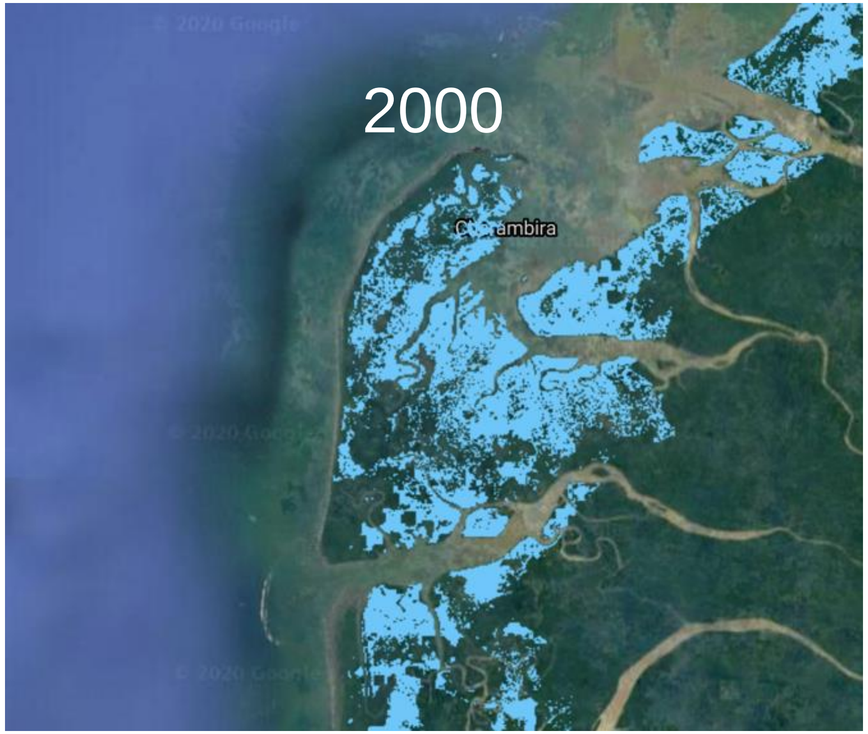
Understanding Change Over Time

- We can use a time series analysis to understand how mangroves have changed over time
- Identify areas of loss/gain
- Understand patterns of change

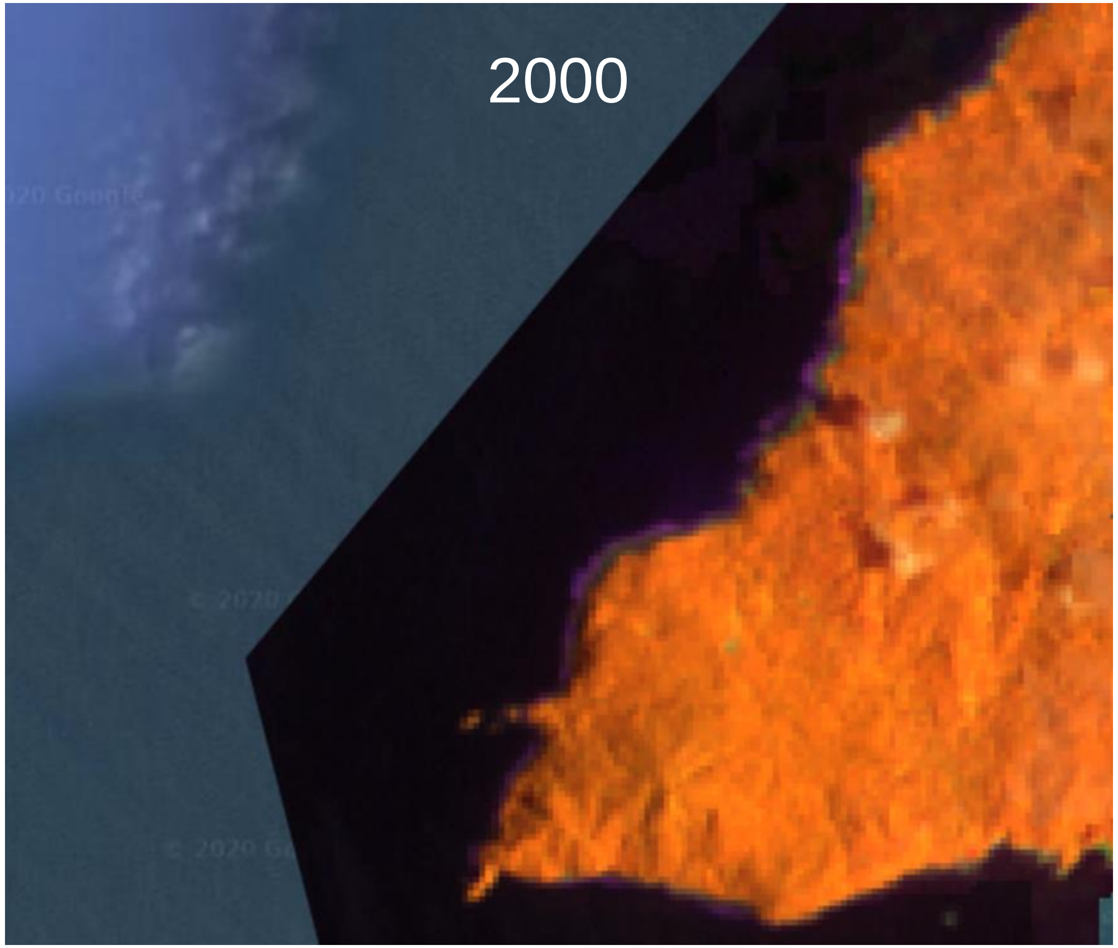


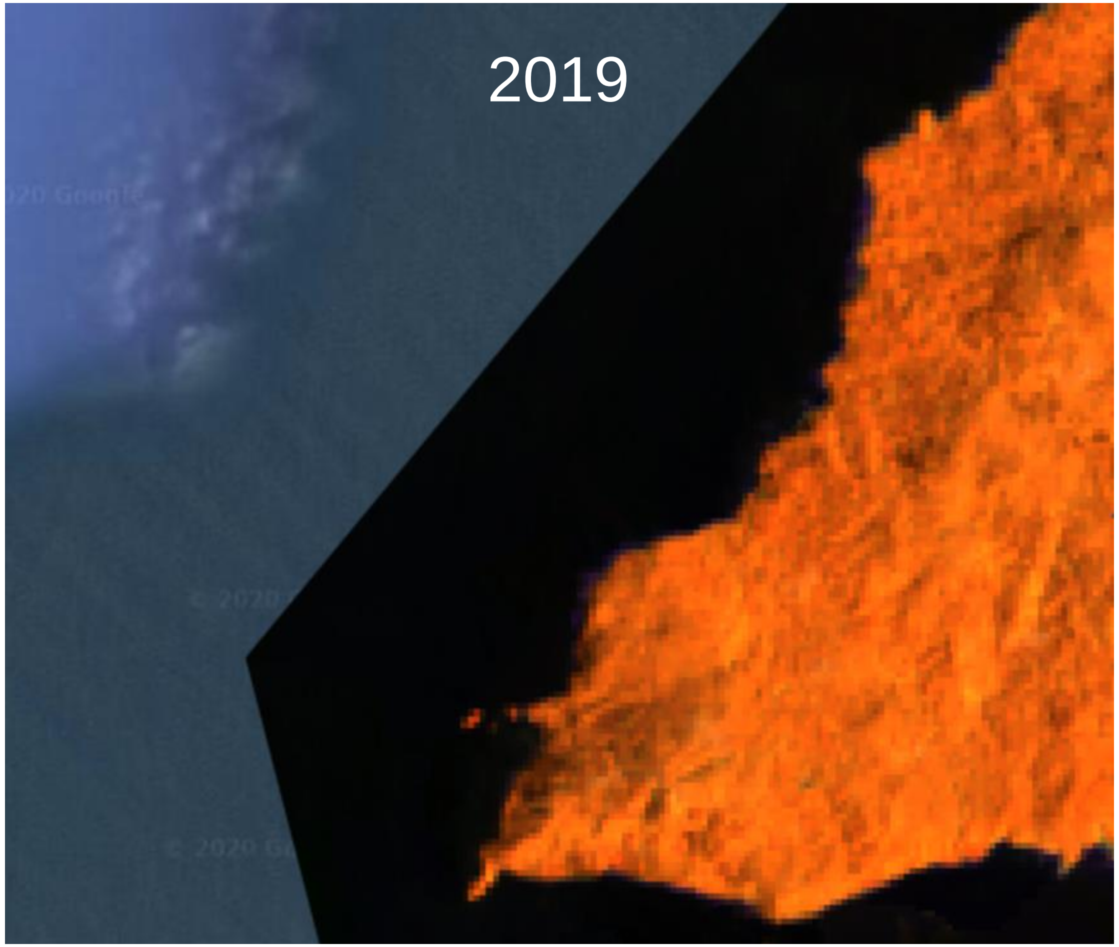
Time Series Analyses

- Examine the same study area over several years
- Focus on specific years of interest (ex: decadal info)
- Example: Comparison of mangrove extent in Colombia, 2000 vs. 2019



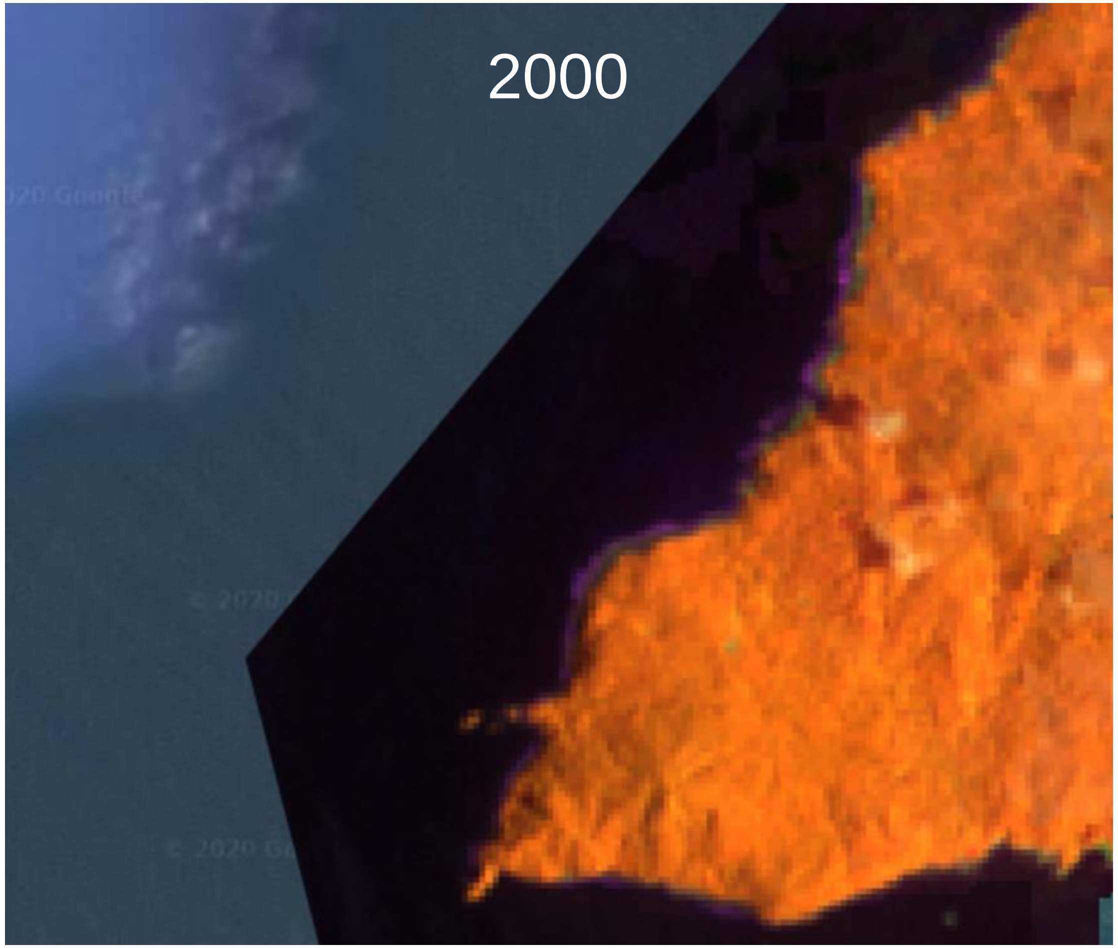
Using Landsat to Compare Extent





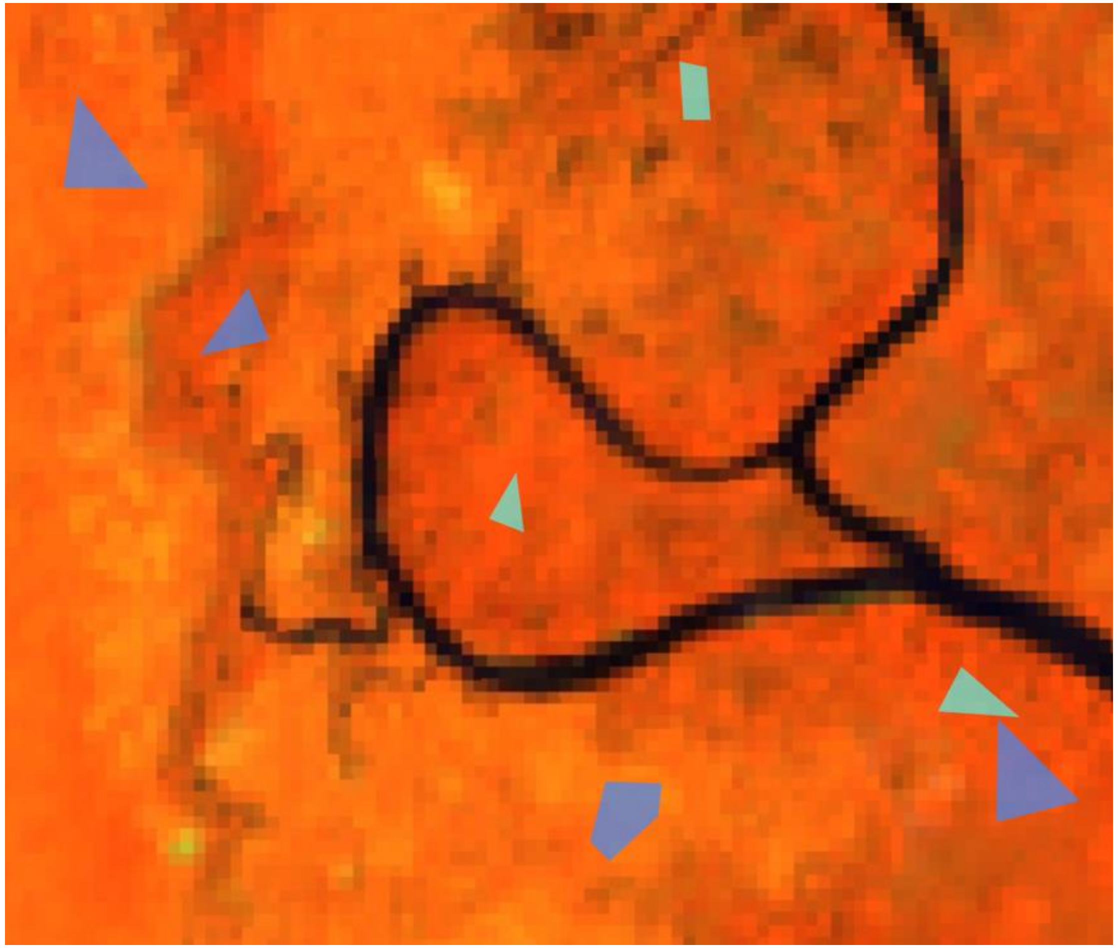
Using Landsat to Compare Extent

- We can compare the values of different indices like Normalized Difference Vegetation Index (NDVI) across different years.
- Higher values of of NDVI indicate higher levels of vegetation (mangroves).



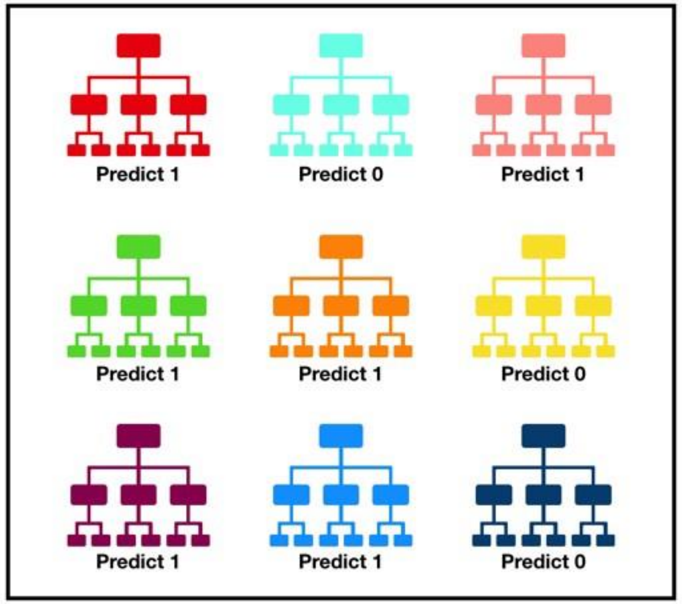
Using Landsat to Compare Extent

- We can create samples of areas with and without mangroves using this imagery.
- Machine learning allows us to use these samples to detect mangroves across a region.



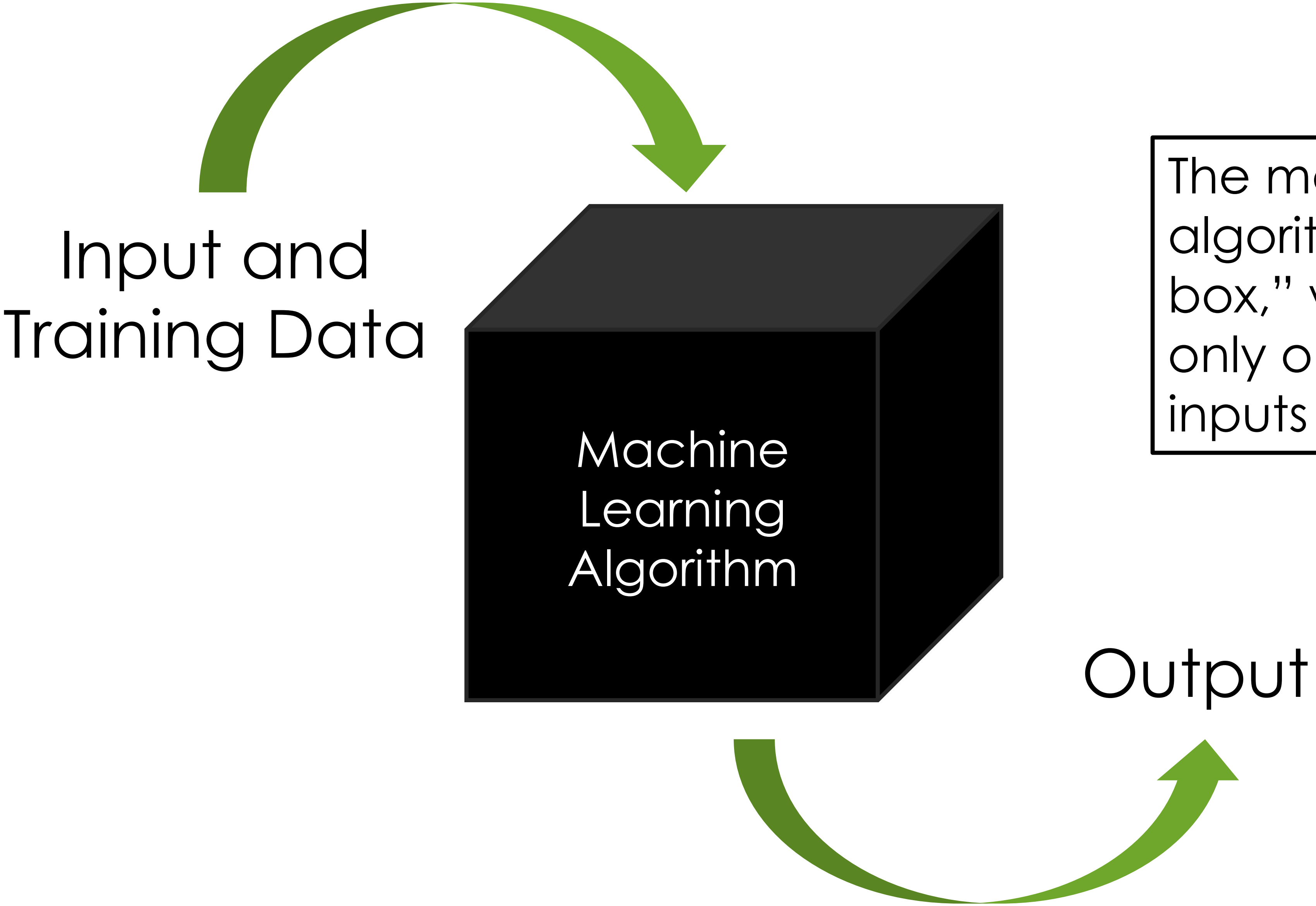
Random Forest Classification

- Machine Learning
  - Uses statistics to identify patterns in large datasets
  - AI that "learns" from data
- Ensemble, tree-based Machine Learning Algorithm
- Supervised
- Uses decision trees to select best solution by "voting"



Tally: Six 1s and Three 0s
Prediction: 1

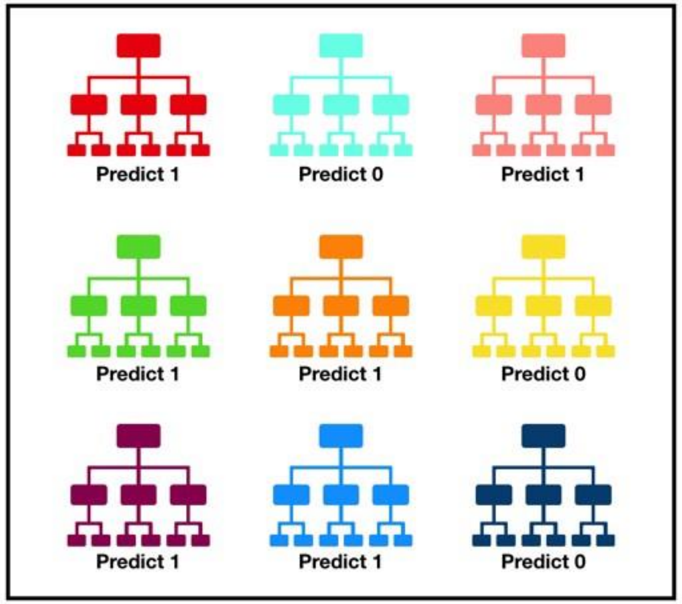
The Black Box



The machine learning algorithm is a "black box," where we can only observe the inputs and outputs.

Random Forest Classification

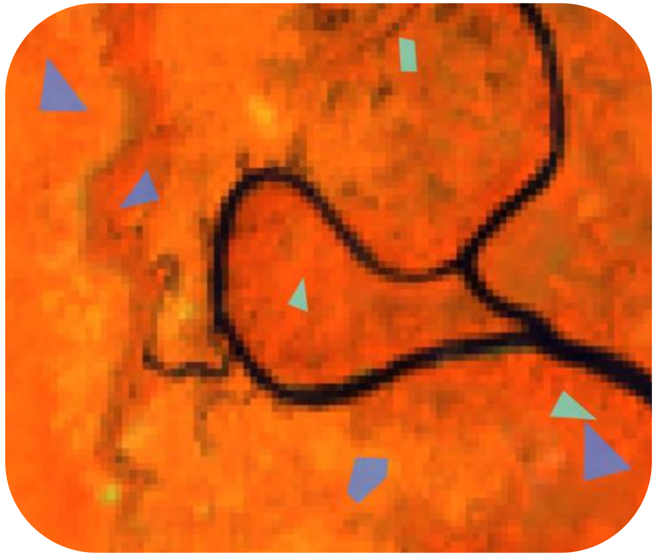
- The algorithm constructs a decision tree for each sample.
- Based on the predictors (bands from Landsat), the trees will vote for each pixel to detect mangrove vs. non-mangrove.
- The most supported value is assigned to each pixel.



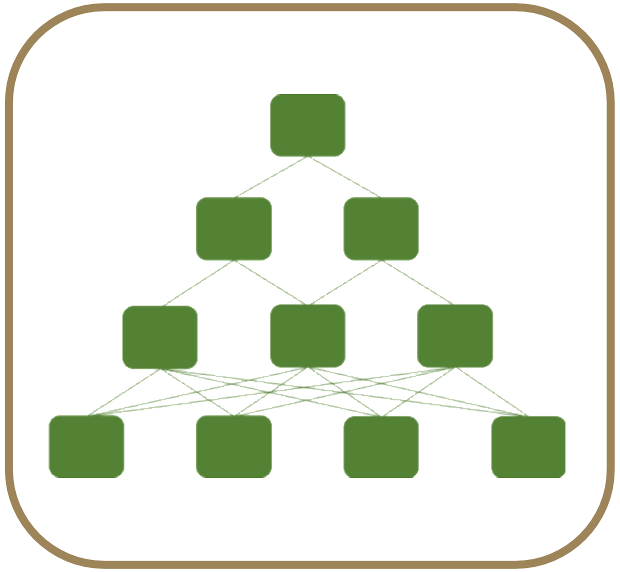
Tally: Six 1s and Three 0s
Prediction: 1

Mangrove Extent Using Random Forest Classification

1) Create Samples



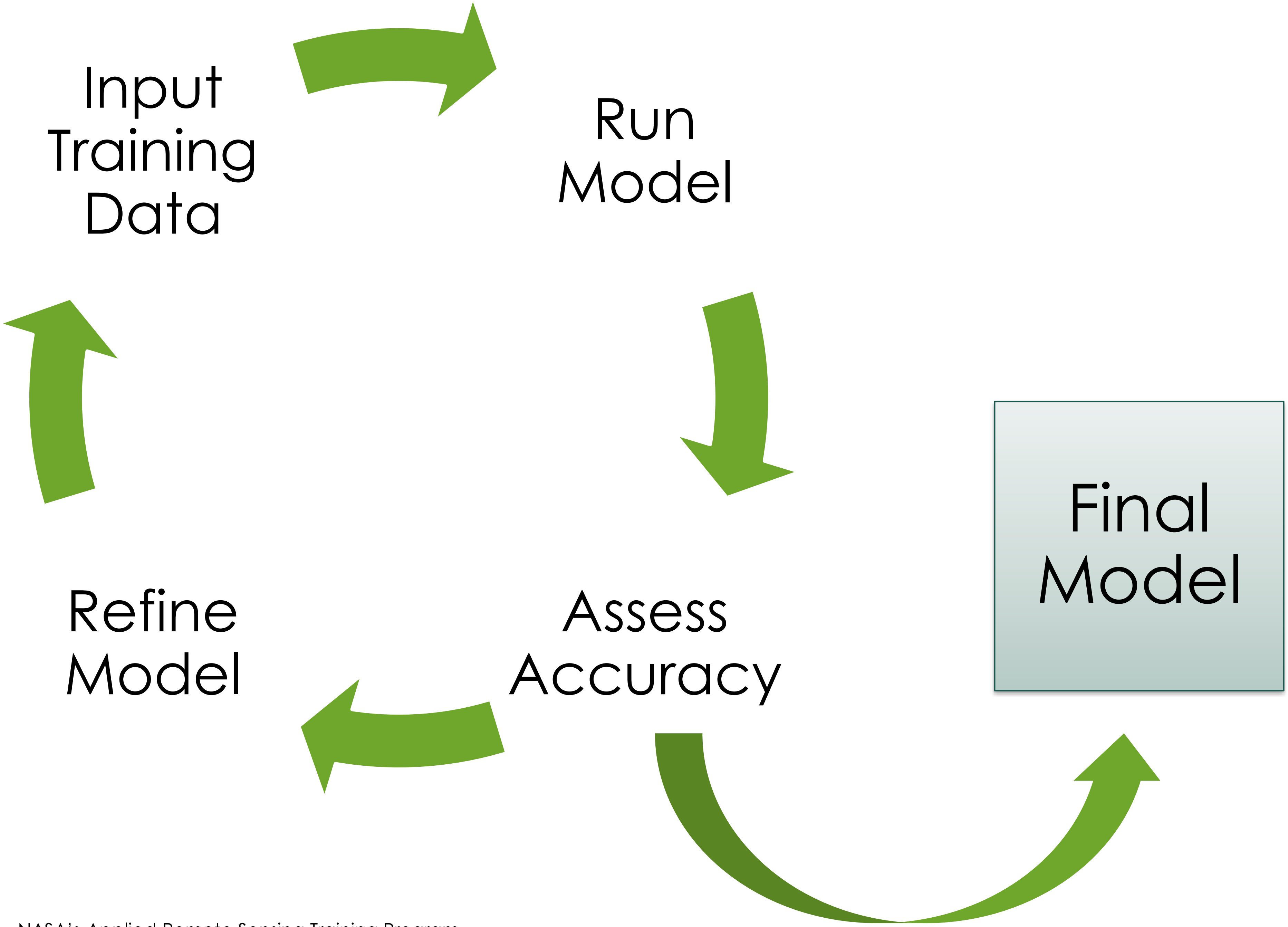
2) Run RFC



3) Refine Model

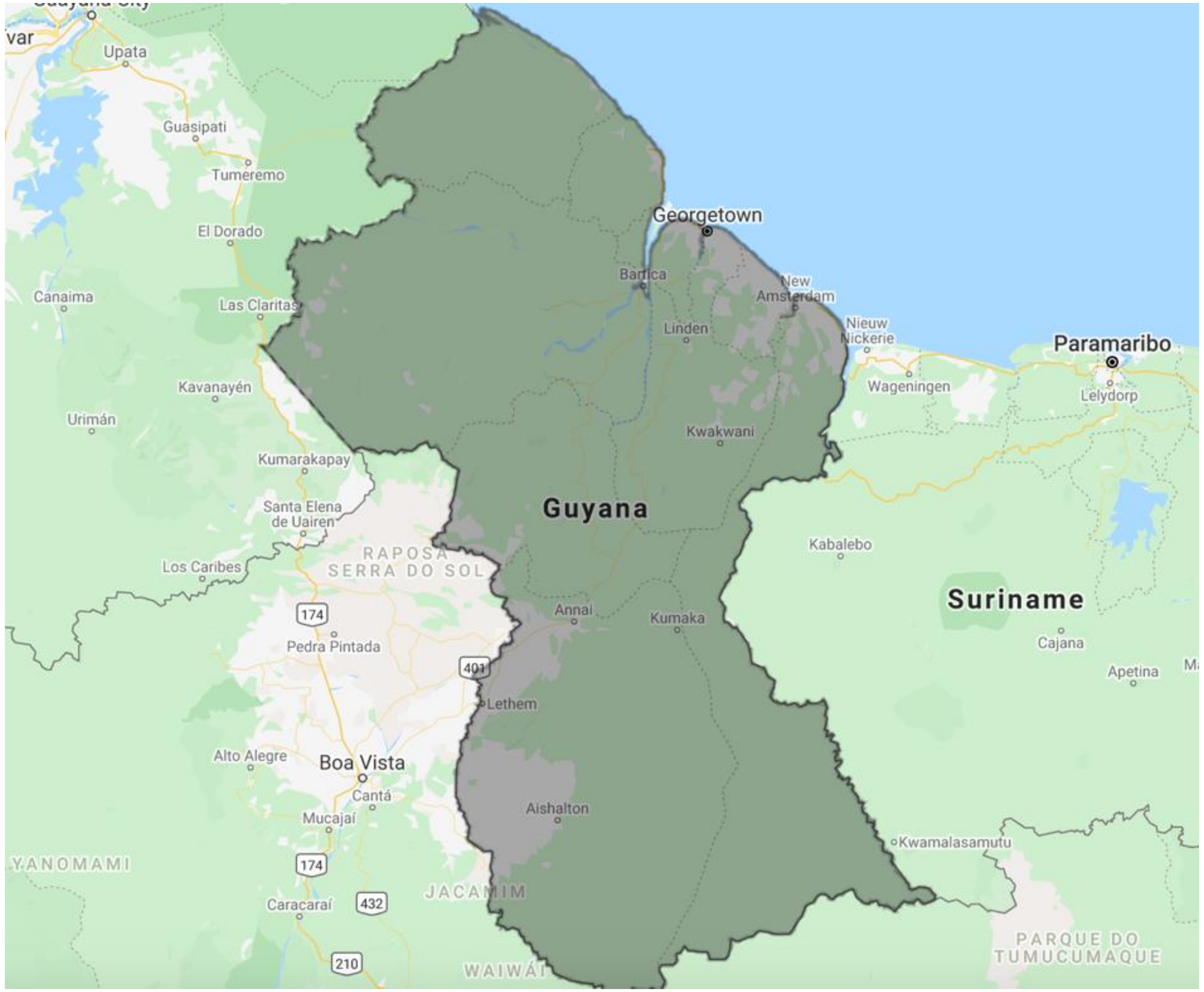
Validation of Results

- After we have created our map, we need to validate it.
- Understanding the accuracy of our model allows us to understand how reliable our results are.
- We can use stratified random samples to run an independent accuracy assessment.
- We ”visit” each point and use satellite imagery to mark if they are correct.
- For this exercise, we will use the Class Accuracy Plug-in in QGIS 3.10:
  - https://github.com/remotesensinginfo/classaccuracy


Input Training Data
Run Model
Assess Accuracy
Refine Model
Final Model

Study Area: Guyana

- NASA and SERVIR work with in-country partners in countries like Guyana to help them monitor mangroves.
- Guyana foresees future flooding and saltwater intrusion as sea levels rise.
- This case study shows how mapping mangroves can help focus conservation practices.



TO FOLLOW ALONG WITH THE FULL SCRIPT:

https://code.earthengine.google.com/a2400e2ce048914ccf1b16aba2702951

1) Setting Up the Map

Import the Following

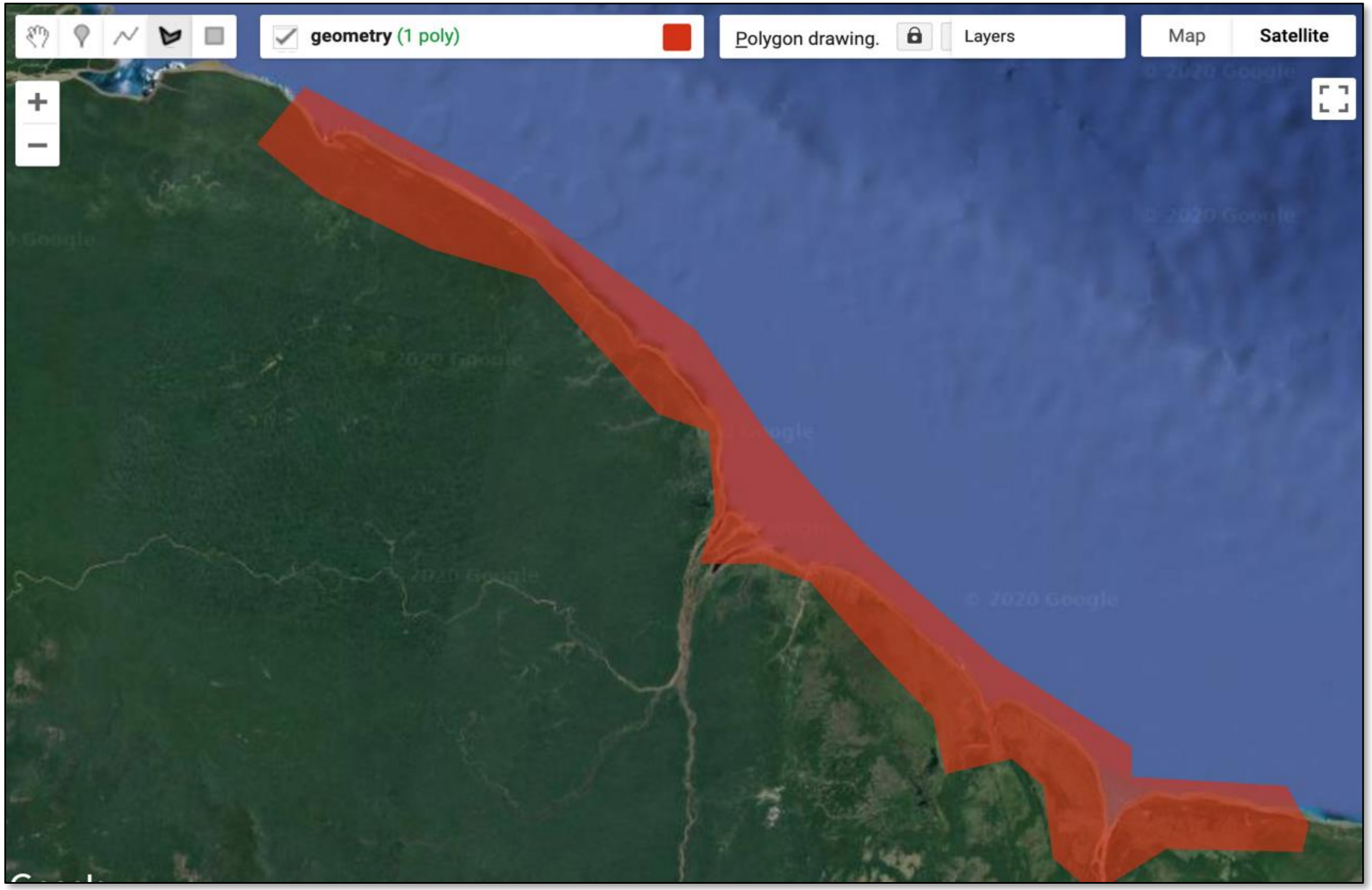
- Landsat 8 Surface Reflectance Tier 1
    - Image Collection: LANDSAT/LC08/C01/T1_SR
    - Rename "L8"
- SRTM Digital Elevation Data 30m
    - Image Collection: USGS/SRTMGL1_003
    - Rename "SRTM"

Start by Setting Up the Map

Center the map to the region of interest using the region shapefile.

```
Map.centerObject(geometry,7)
Map.setOptions('satellite')
```

Draw a Geometry for Area of Interest



2) Setting Up a Filtered Landsat Composite

Set Up a Filtered Landsat Composite

First, we need to mask clouds.

Landsat data includes a 'pixel_qa' band which can be used to create a function to mask clouds.

```
function maskClouds(image) {
        // Bits 3 and 5 are cloud shadow and cloud, respectively.
        var cloudShadowBitMask = ee.Number(2).pow(3).int();
        var cloudsBitMask = ee.Number(2).pow(5).int();
        // Get the pixel QA band.
        var qa = image.select('pixel_qa');
         // Both flags should be set to zero, indicating clear conditions.
        var mask = qa.bitwiseAnd(cloudShadowBitMask).eq(0).and(qa.bitwiseAnd(cloudsBitMask).eq(0));
         // Return the masked image, scaled to [0, 1].
 return image.updateMask(mask).divide(10000).copyProperties(image, ["system:time_start"]);
}
```

Spectral Indices

- **NDVI = Normalized Difference Vegetation Index (Red and NIR)**
  - Quantifies Vegetation
- **NDMI = Normalized Difference Moisture Index (NIR and SWIR)**
  - Vegetation Water Content
- **MNDWI = Modified Normalized Difference Water Index (Green and SWIR)**
  - Water Information
- **SR = Simple Ratio (Red and NIR)**
  - Simple Vegetation Index
- **Ratio54 = Band Ratio 54 (SWIR and NIR)**
  - Maps Water Features
- **Ratio35 = Band Ratio 35 (Red and SWIR)**
  - Maps Water Features
- **GCVI = Green Chlorophyll Vegetation Index (NIR and Green)**
  - Green Leaf Biomass

Add Spectral Indices

This function maps spectral indices for Mangrove Mapping using Landsat 8.

```
var addIndicesL8 = function(img) {
    var ndvi = img.normalizedDifference(['B5','B4']).rename('NDVI'); // NDVI
    // NDMI (Normalized Difference Mangrove Index - Shi et al 2016 - New spectral metrics for mangrove forest identification)
    var ndmi = img.normalizedDifference(['B7','B3']).rename('NDMI');
    var mndwi = img.normalizedDifference(['B3','B6']).rename('MNDWI'); // MNDWI (Modified Normalized Difference Water ndex - Hanqiu Xu, 2006)
    var sr = img.select('B5').divide(img.select('B4')).rename('SR'); // SR (Simple Ratio)
    var ratio54 = img.select('B6').divide(img.select('B5')).rename('R54'); // Band Ratio 54
    var ratio35 = img.select('B4').divide(img.select('B6')).rename('R35'); // Band Ratio 35
    var gcvi = img.expression('(NIR/GREEN)-1',{ 'NIR':img.select('B5'), 'GREEN':img.select('B3')}).rename('GCVI'); // GCVI
    return img
    .addBands(ndvi)
    .addBands(ndmi)
    .addBands(mndwi)
    .addBands(sr)
    .addBands(ratio54)
    .addBands(ratio35)
    .addBands(gcvi);
};
```

Filter Landsat Data by Date

```
// Select the desired central year here
var year = 2019;

// Start date will be set one year before the central year
var startDate = (year-1)+'-01-01';

// End date will be set to one year later than the central year
var endDate = (year+1)+'-12-31';
```

Apply Filters and Masks to Landsat 8 Imagery

```
var l8 = L8.filterDate(startDate,endDate)
// Mask for clouds and cloud shadows
    .map(maskClouds)
// Add the indices
    .map(addIndicesL8)
```

Composite the Landsat Image Collection

You can composite on a per-pixel, per-band basis using ".median()"

```
var composite = l8
      // Uses the median reducer
      .median()
      // Clips the composite to our area of interest
      .clip(geometry);
```

Mask to Areas of Low Elevation and High NDVI and MNDWI

Additional masking allows us to focus on areas that are more likely to have mangroves.

```
// Clip SRTM data to region
        var srtmClip = SRTM.clip(geometry);
// Mask to elevations less than 65 meters
        var elevationMask = srtmClip.lt(65);
// Used the NDVI and MNDWI bands to create masks
        var NDVIMask = composite.select('NDVI').gt(0.25);
        var MNDWIMask = composite.select('MNDWI').gt(-0.50);
// Apply the masks
        var compositeNew = composite
            .updateMask(NDVIMask)
            .updateMask(MNDWIMask)
            .updateMask(elevationMask)
```

Display Results

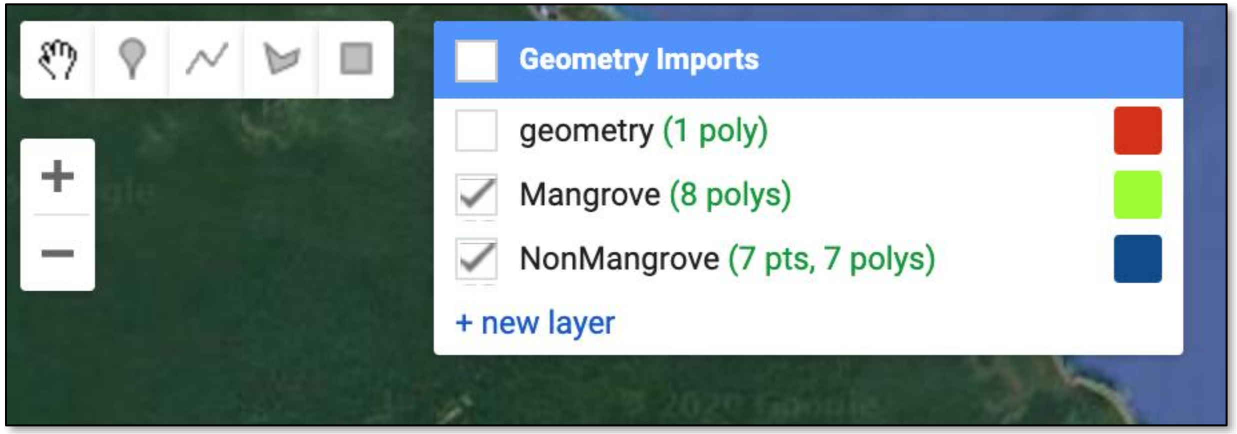
We need to map the Landsat composite to assemble our training data.

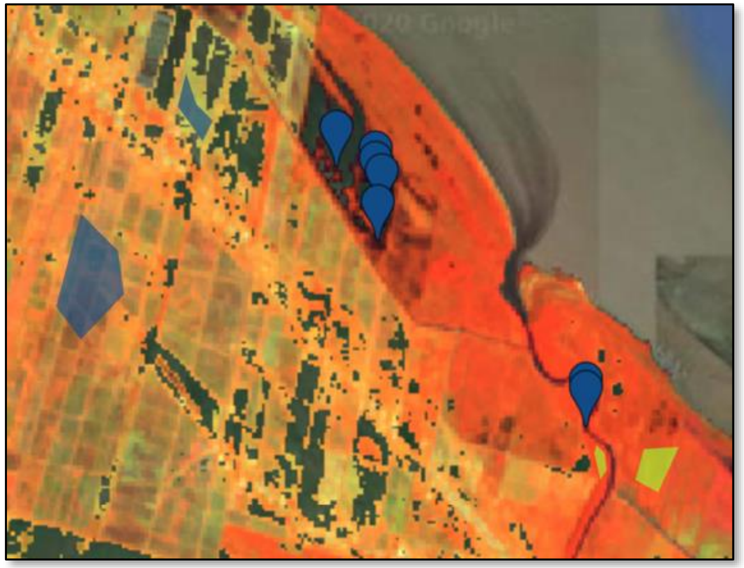
```
// Select bands and parameters for visualization
var visPar = {bands:['B5','B6','B4'], min: 0, max: 0.35};

// Add layer to map
Map.addLayer(compositeNew
        .clip(geometry), visPar, 'Landsat
        Composite 2019')
```

3) Constructing the Model

Construct a Random Forest Classification

Use the displayed Landsat data to add polygons of training data.





Prepare Training Data and Predictors

```
// After drawing training polygons, merge them together
var classes = Mangrove.merge(NonMangrove)

// Define the bands you want to include in the model
var bands = ['B5','B6','B4','NDVI','MNDWI','SR','GCVI']

// Create a variable called image to select the bands of interest and clip to geometry
var image = compositeNew.select(bands).clip(geometry)

// Assemble samples for the model
var samples = image.sampleRegions({
  collection: classes, // Set of geometries selected for training
  properties: ['landcover'], // Label from each geometry
  scale: 30 // Make each sample the same size as Landsat pixel
  }).randomColumn('random'); // creates a column with random numbers
```

Split Samples for Testing

Here we randomly split our samples to set some aside for testing our model's accuracy using the "random" column we created.

```
var split = 0.8; // Roughly 80% for training, 20% for testing
var training = samples.filter(ee.Filter.lt('random', split)); // Subset training data
var testing = samples.filter(ee.Filter.gte('random', split)); // Subset testing data
```

Split Samples for Testing

Print these variables to see how much training and testing data you are using.

```
print('Samples n =', samples.aggregate_count('.all'));
print('Training n =', training.aggregate_count('.all'));
print('Testing n =', testing.aggregate_count('.all'));
```

```
Samples n =                JSON
8890

Training n =               JSON
7167

"Testing n ="              Formatted
1723
```

Begin Random Forest Classification

".smileRandomForest" is used to run the model. Here we run the model using 100 trees and 5 randomly selected predictors per split ("(100,5)").

```
var classifier = ee.Classifier.smileRandomForest(100,5).train({
  features: training.select(['B5','B6','B4','NDVI','MNDWI','SR','GCVI',
        'landcover']), // Train using bands and landcover property
  classProperty: 'landcover', // Pull the landcover property from
                              // classes
  inputProperties: bands
  });
```

Test the Fit of the Model

```
var validation = testing.classify(classifier);
var testAccuracy = validation.errorMatrix('landcover’, 'classification');
 print('Validation error matrix RF: ', testAccuracy);
 print('Validation overall accuracy RF: ‘,

testAccuracy.accuracy());
```

```
Validation error matrix RF:                JSON
[[1295,0],[0,428]]                         JSON

Validation overall accuracy RF:            JSON
1
```

Classify the Landsat Composite Using the Random Forest Model

```
var classifiedrf = image.select(bands) // select the predictors
                   .classify(classifier);
                   // .classify applies the Random Forest
```

Reduce Noise in Results

The model results may be "noisy." To reduce noise, create a mask to mask unconnected pixels.

```
// Create an image that shows the number of pixels each
// pixel is connected to
var pixelcount = classifiedrf.connectedPixelCount(100, false);
// Filter out all pixels connected to 4 or less
var countmask = pixelcount.select(0).gt(25);
```

Mask Results

Mask the results to only display mangrove extent.

```
var classMask = classifiedrf.select('classification').gt(0)
var classed = classifiedrf.updateMask(countmask)
                          .updateMask(classMask)
```

Map Results

Mask the results to only display mangrove extent

```
// Add classification to map
Map.addLayer (classed, {min: 1, max: 1, palette:'blue'}, 'Mangrove Extent 2019');

// For comparison, let's add the GMW dataset to the map.
var GMW = ee.Image('projects/mangrovescience/GuyanaGMW')
Map.addLayer (GMW, {min: 1, max: 1, palette:'green'}, 'Global Mangrove Watch');
```

4) Time Series Comparison

Time Series Comparison

We want to be able to compare mangrove extent in different years to examine if mangrove area has been lost over time.

We need to rerun our model using Landsat imagery from a different date of interest with new training data.

Adding Landsat 7 Spectral Indices

- Landsat 5 and 7 have different band numbers.
- Landsat 8 also collects images in wider swaths.
- We will need to assign spectral indices using different band values.

Spectral Indices for Landsat 8

Table 1. Landsat 8 Operational Land Imager Spectral bands

Bands	Wavelength (µm)	Resolution (m)
Band 1 - Coastal aerosol	0.43-0.45	30
Band 2 - Blue	0.45-0.51	30
Band 3 - Green	0.53-0.59	30
Band 4 - Red	0.64-0.67	30
Band 5 - Near Infrared (NIR)	0.85-0.88	30
Band 6 - SWIR 1	1.57-1.65	30
Band 7 - SWIR 2	2.11-2.29	30
Band 8 - Panchromatic	0.50-0.68	15
Band 9 - Cirrus	1.36-1.38	30
Band 10 - Thermal Infrared (TIRS) 1	10.6-11.19	100
Band 11 - Thermal Infrared (TIRS) 2	11.50-12.51	100

Spectral Indices for Landsat 7

Table 2. Landsat 7 Enhanced Thematic Mapper+ Spectral bands

Bands	Wavelength (µm)	Resolution (m)
Band 1	0.45-0.52	30
Band 2	0.52-0.60	30
Band 3	0.63-0.69	30
Band 4	0.77-0.90	30
Band 5	1.55-1.75	30
Band 6	10.40-12.50	60 (30)
Band 7	2.09-2.35	30
Band 8	.52-.90	15
Band 1	0.45-0.52	30
Band 2	0.52-0.60	30
Band 3	0.63-0.69	30

Adding Landsat 7 Spectral Indices

```
var addIndicesL7 = function(img) {
var ndvi = img.normalizedDifference(['B4','B3']).rename('NDVI');
var ndmi = img.normalizedDifference(['B7','B2']).rename('NDMI');
var mndwi = img.normalizedDifference(['B2','B5']).rename('MNDWI');
var sr = img.select('B4').divide(img.select('B3')).rename('SR');
var ratio54 = img.select('B5').divide(img.select('B4')).rename('R54');
var ratio35 = img.select('B3').divide(img.select('B5')).rename('R35');
var gcvi = img.expression('(NIR/GREEN)-1',{ 'NIR':img.select('B4'),  'GREEN':img.select('B2')}).rename('GCVI');
  return img.addBands(ndvi)
    .addBands(ndmi)
    .addBands(mndwi)
    .addBands(sr)
    .addBands(ratio54)
    .addBands(ratio35)
    .addBands(gcvi)};
```

Filter Landsat Data by Date and Region

```
// Select the desired central year here
var year = 2009;

// Start date will be set one year before the central year
var startDate = (year-1)+'-01-01';

// End date will be set to one year later than the central year
var endDate = (year+1)+'-12-31';
```

Apply Filters and Masks

```
var l7 = L7.filterDate(startDate,endDate)
// Mask for clouds and cloud shadows
// We use the same function we used for Landsat 8 to mask clouds
   .map(maskClouds)
// Add the indices
   .map(addIndicesL7)
```

Composite Image Collection

```
var L7composite = l7
        // Uses the median reducer
        .median()
        // Clips the composite to our area of interest
        .clip(geometry);
```

Mask to Low Elevation/High NDVI and MNDWI

```
var L7NDVIMask = L7composite.select('NDVI').gt(0.25);
var L7MNDWIMask = L7composite.select('MNDWI').gt(-0.50);
// Apply the masks
var L7compositeNew = L7composite
                .updateMask(L7NDVIMask)
                .updateMask(L7MNDWIMask)
                .updateMask(elevationMask) //We use the same mask
```

Display Results

```
// Select bands and parameters for visualization
// We use bands 4, 5, and 3 instead
var L7visPar = {bands:['B4','B5','B3'], min: 0, max: 0.35};

// Add layer to map
Map.addLayer(L7compositeNew.clip(geometry), L7visPar, 'Landsat Composite 2009')
```

5) New Random Forest Model

Construct New Random Forest Model

Prepare training data and predictors.

```
// After drawing training polygons, merge them together.
var classes2009 = Mangrove2009.merge(NonMangrove2009)

// Define the bands you want to include in the model.
var L7bands = ['B4','B5','B3','NDVI','MNDWI','SR','GCVI']

// Create a variable called “image” to select the bands of interest and clip to geometry.
var L7image = L7compositeNew.select(L7bands).clip(geometry)
```

Assemble Samples

```
var L7samples = L7image.sampleRegions({
  collection: classes2009, // Set of geometries selected for training
  properties: ['landcover'], // Label from each geometry
  scale: 30 // Make each sample the same size as Landsat pixel
  }).randomColumn('random'); // creates a column with random numbers
```

Split Samples for Testing

```
// Subset training data.
var L7training = L7samples.filter(ee.Filter.lt('random', split));

// Subset testing data.
var L7testing = L7samples.filter(ee.Filter.gte('random', split));

// Print these variables to see how much training and testing data you are
// using.
  print('Samples n =', L7samples.aggregate_count('.all'));
  print('Training n =', L7training.aggregate_count('.all'));
  print('Testing n =', L7testing.aggregate_count('.all'));
```

Begin Random Forest Classification

```
var L7classifier = ee.Classifier.smileRandomForest(100,5).train({
  features: L7training.select(['B4','B5','B3','NDVI','MNDWI','SR','GCVI',
'landcover']), // Train using bands and landcover property
  classProperty: 'landcover', // Pull the landcover property from classes
  inputProperties: L7bands
  });
```

Classify Landsat Composite

```
var L7classifiedrf = L7image.select(L7bands) // Select the predictors
                     .classify(L7classifier);

// Reduce Noise
var pixelcount = L7classifiedrf.connectedPixelCount(100, false);
var countmask = pixelcount.select(0).gt(25);
```

Map Results

```
// Mask results to only display mangrove extent
    var L7classMask = L7classifiedrf.select('classification').gt(0)
    var L7classed=
L7classifiedrf.updateMask(countmask).updateMask(L7classMask)

Map.addLayer (L7classed, {min: 1, max: 1, palette:'green'}, 'Mangrove Extent
2009');
```


© 2020 Google

6) Calculate Mangrove Area

Calculate Mangrove Extent

```
// 2009
var get2009 = L7classed.multiply(ee.Image.pixelArea()).divide(10000).reduceRegion({
    reducer:ee.Reducer.sum(),
    geometry:geometry,
    scale: 100,
    maxPixels:1e13,
    tileScale: 16
    }).get('classification');

print(get2009, 'Mangrove Extent 2009 in ha')
```

Calculate Mangrove Extent

```
// 2019
var get2019 = classed.multiply(ee.Image.pixelArea()).divide(10000).reduceRegion({
    reducer:ee.Reducer.sum(),
    geometry:geometry,
    scale: 100,
    maxPixels:1e13,
    tileScale: 16
    }).get('classification');

print(get2019, 'Mangrove Extent 2019 in ha')
```

7) Create Points for Accuracy Assessment

Running Independent Accuracy Assessments

To test the actual accuracy of the model rather than the fit, we will need to create random sampling points.

```
var stratSamples = classed.stratifiedSample({
        numPoints:150,      // Number of points per class
        classBand: 'classification',
        region:geometry,
        scale: 30,
        geometries:true });
```

Buffer Points

```
// Add a 15m-radius buffer around each point.
var stratBuff = function(feature) {
    var num = feature.get('classification');
    return feature.buffer(15).set('classification', num);
    };

var stratPoints = stratSamples.map(stratBuff)
```

8) Export Layers of Interest

Export Layers of Interest

```
// 2019 Mangrove Extent
Export.image.toDrive({
  image: classed,
  description: '2019GuyanaMangroveExtent',
  region: geometry,
  scale: 30,
  maxPixels: 1e13
  });
```

Export Layers of Interest

```
// 2009 Mangrove Extent
Export.image.toDrive({
  image: L7classed,
  description: '2009GuyanaMangroveExtent',
  region: geometry,
  scale: 30,
  maxPixels: 1e13
  });
```

Export Layers of Interest

```
// Stratified Random Samples
Export.table.toDrive({
  collection: stratPoints,
  description:'StratifiedrandomPoints',
  fileFormat: 'SHP',
});
```

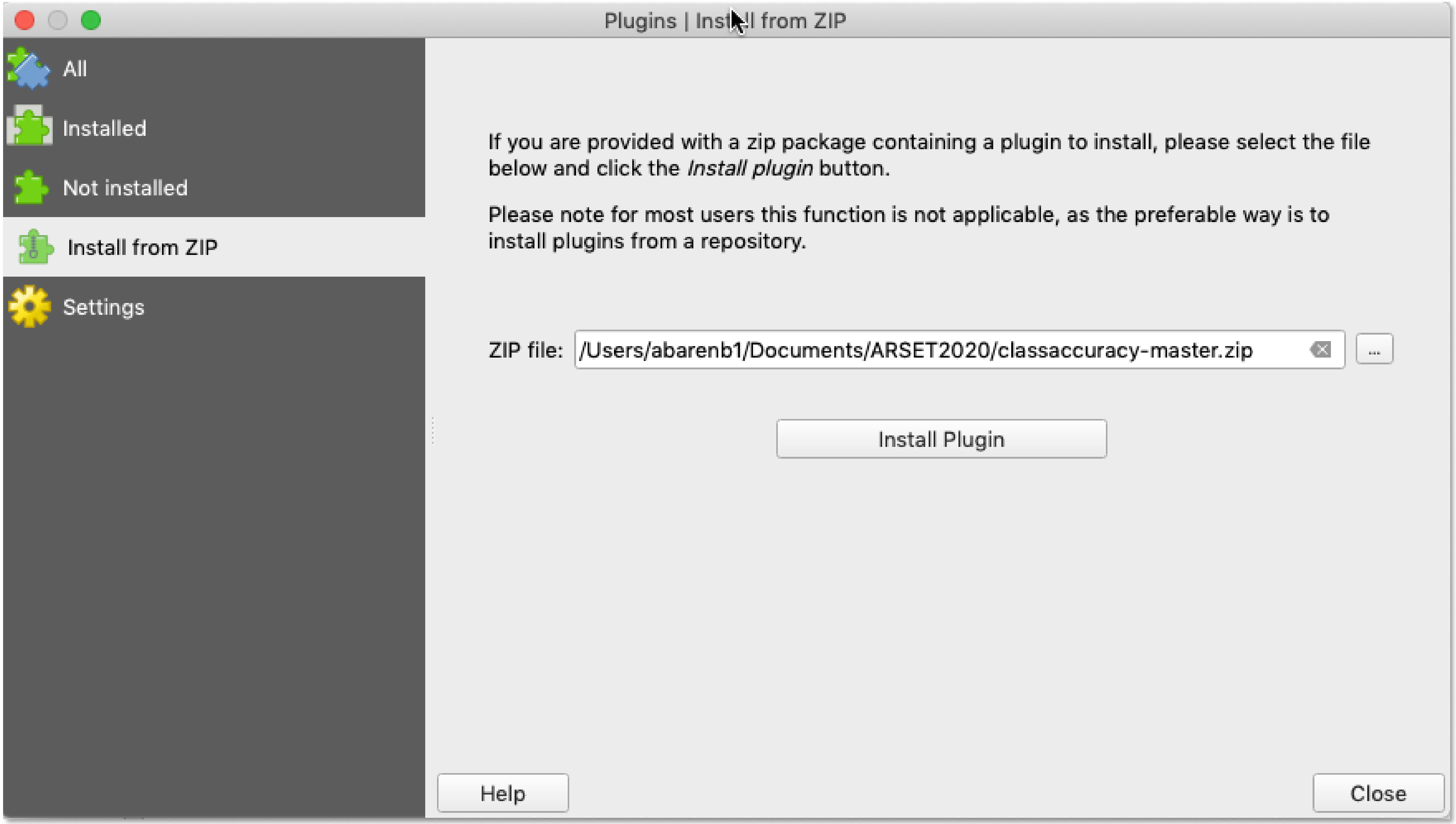
9) QGIS Class Accuracy

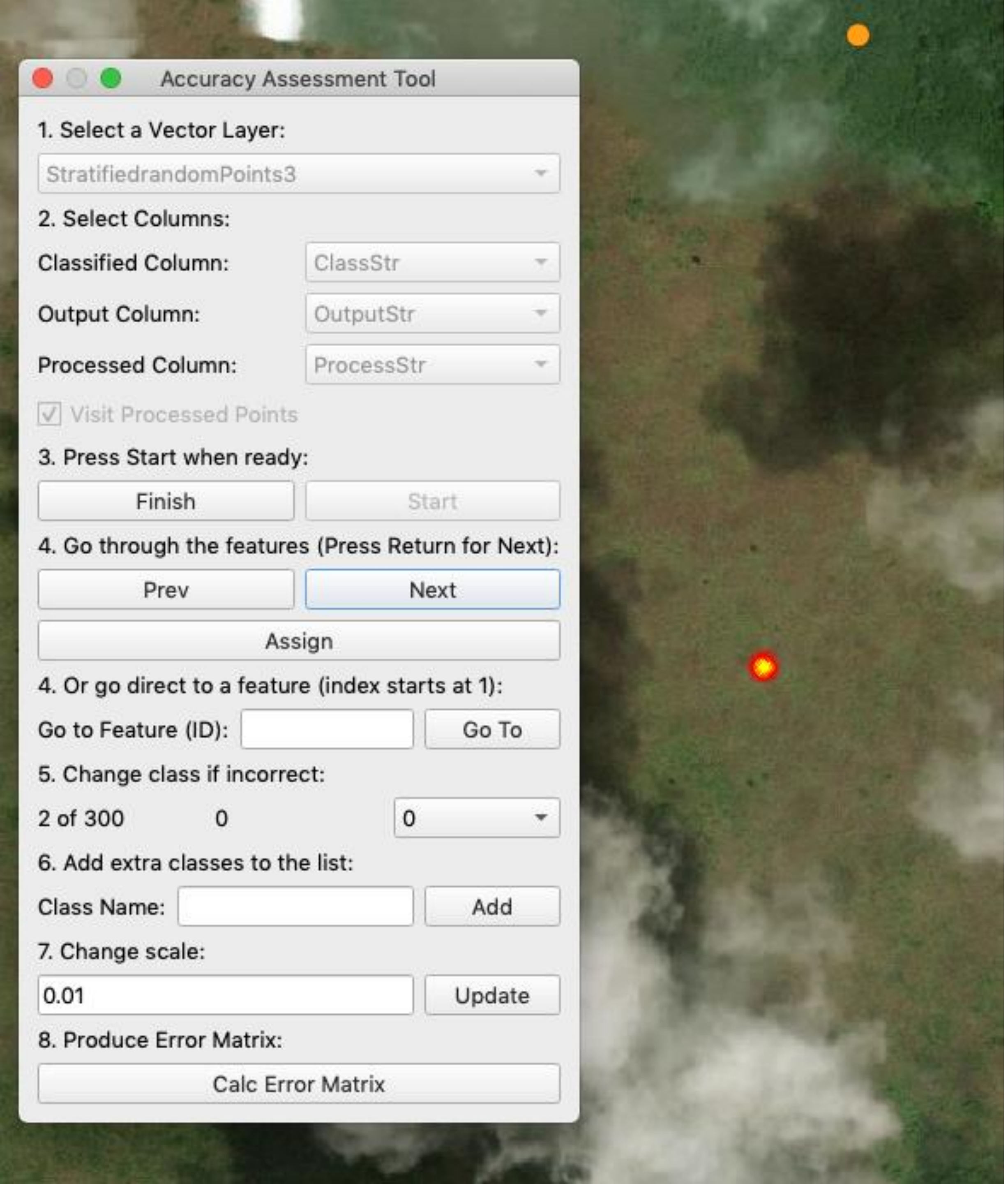
Class Accuracy Plug-in

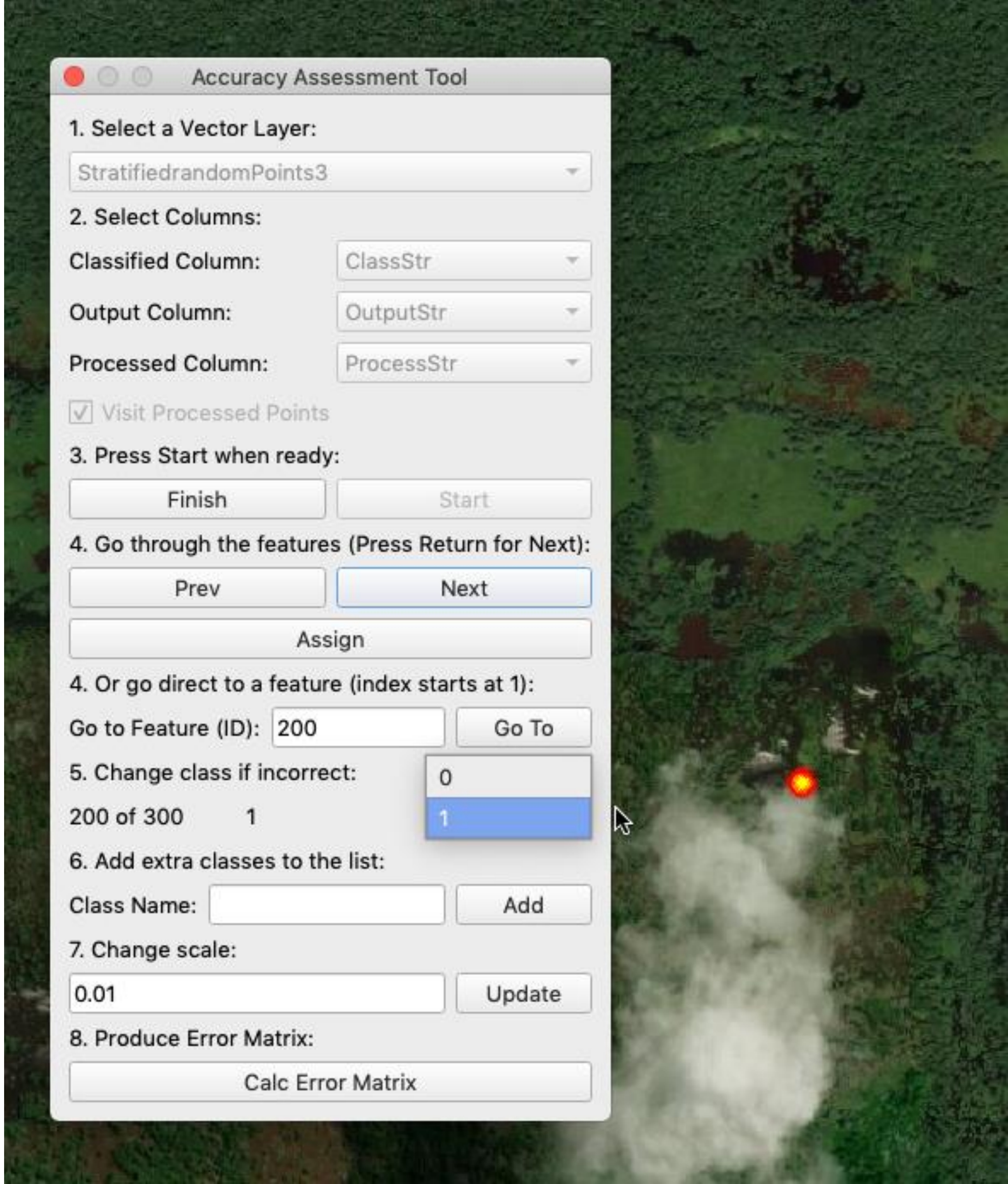
- Class Accuracy is a QGIS Plug-in created by Dr. Peter Bunting.
- This tool takes the user through each randomly stratified point.
- The user then defines whether the point was accurately classed by the model or not.
- The result is an assessment of the overall accuracy of the model.

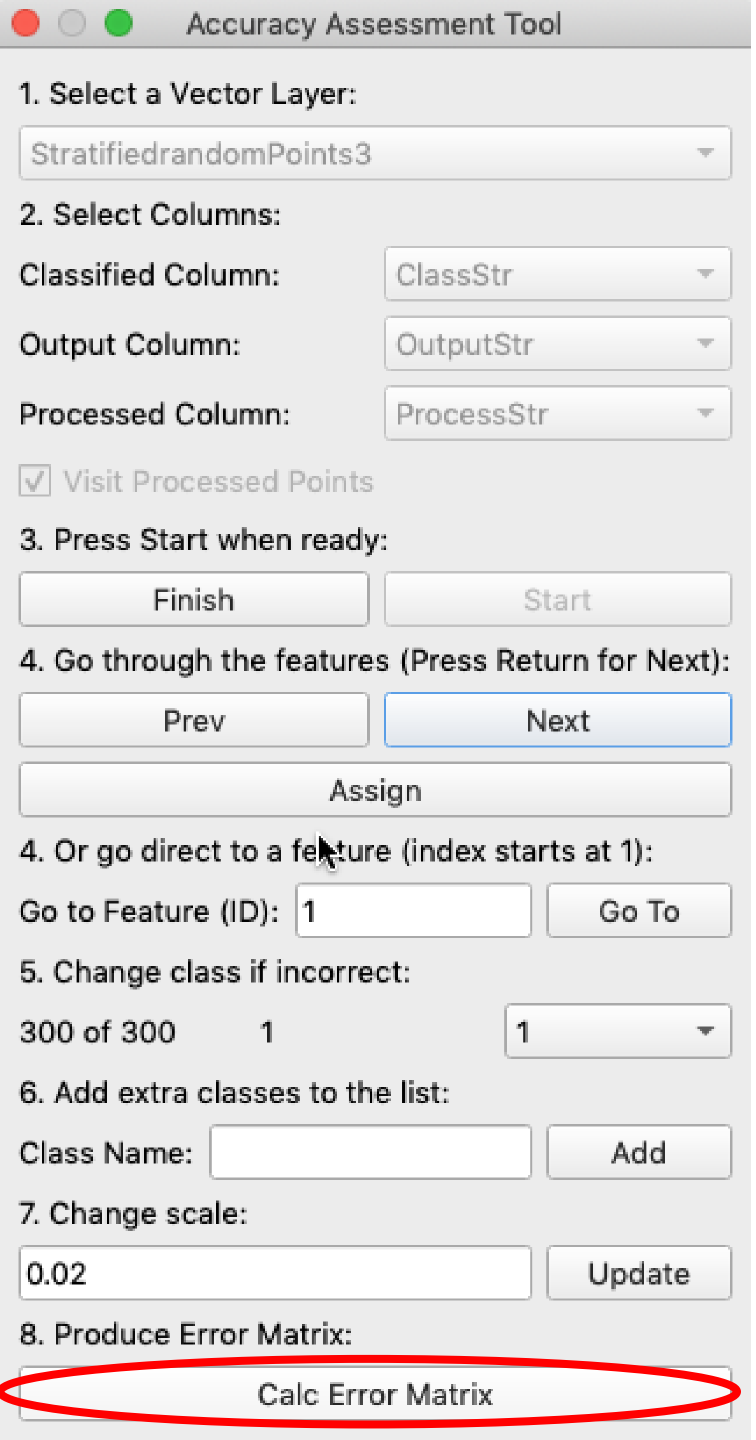
QGIS Class Accuracy Plug-In

- Open QGIS 3.10 and add a satellite basemap (ex. Bing Aerial).
- Add exported random points:
  - Add two columns: Export and Processed
  - Ensure all columns including classification are in **String** format.
- Open Class Accuracy Plug-in:
  - https://github.com/remotesensinginfo/classaccuracy


Plugins | Install from ZIP
All
Installed
Not installed
Install from ZIP
Settings
If you are provided with a zip package containing a plugin to install, please select the file below and click the *Install plugin* button.
Please note for most users this function is not applicable, as the preferable way is to install plugins from a repository.
ZIP file: /Users/abarenb1/Documents/ARSET2020/classaccuracy-master.zip
...
Install Plugin
Help
Close


Accuracy Assessment Tool
1. Select a Vector Layer:
StratifiedrandomPoints3
2. Select Columns:
Classified Column:
ClassStr
Output Column:
OutputStr
Processed Column:
ProcessStr
Visit Processed Points
3. Press Start when ready:
Finish
Start
4. Go through the features (Press Return for Next):
Prev
Next
Assign
4. Or go direct to a feature (index starts at 1):
Go to Feature (ID):
Go To
5. Change class if incorrect:
2 of 300
0
0
6. Add extra classes to the list:
Class Name:
Add
7. Change scale:
0.01
Update
8. Produce Error Matrix:
Calc Error Matrix


Accuracy Assessment Tool
1. Select a Vector Layer:
StratifiedrandomPoints3
2. Select Columns:
Classified Column:
ClassStr
Output Column:
OutputStr
Processed Column:
ProcessStr
Visit Processed Points
3. Press Start when ready:
Finish
Start
4. Go through the features (Press Return for Next):
Prev
Next
Assign
4. Or go direct to a feature (index starts at 1):
Go to Feature (ID):
200
Go To
5. Change class if incorrect:
0
200 of 300
1
1
6. Add extra classes to the list:
Class Name:
Add
7. Change scale:
0.01
Update
8. Produce Error Matrix:
Calc Error Matrix



Overall Accuracy (%)	78						
kappa	0.71						
Counts:							
	3	0	2	1	Mixed	NA	User
3	137	6	0	0	3	4	150
0	34	85	8	2	11	10	150
2	0	13	124	6	2	5	150
1	5	11	4	122	4	4	150
Mixed	0	0	0	0	0	0	0
NA	0	0	0	0	0	0	0
Producer	176	115	136	130	20	23	468
Percentage:							
	3	0	2	1	Mixed	NA	User (%)
3	22.83	1	0	0	0.5	0.67	91.33
0	5.67	14.17	1.33	0.33	1.83	1.67	56.67
2	0	2.17	20.67	1	0.33	0.83	82.67
1	0.83	1.83	0.67	20.33	0.67	0.67	81.33
Mixed	0	0	0	0	0	0	0
NA	0	0	0	0	0	0	0
Producer (%)	77.84	73.91	91.18	93.85	0	0	78

Recap

During this lesson we covered:

- Mangrove extent mapping over two time periods
- Independent accuracy assessment of the model

Next time we will cover:

- Creating country-specific apps
- Example applications of results

Questions

- Please enter your questions into the Q&A box.
- We will post the questions and answers to the training website following the conclusion of the course.

Contacts

- Contacts
  – Abigail Barenblitt: abigail.barenblitt@nasa.gov
  – Temilola Fatoyinbo: lola.fatoyinbo@nasa.gov

- General ARSET Inquiries
  – Ana Prados: aprados@umbc.edu

- ARSET Website:
  – appliedsciences.nasa.gov/arset

Thank You!