

It is possible to access to it via:

<https://code.earthengine.google.com/9b0126e1ca88b93a12b8052747f80fbd> or by copy-paste the following code in the GEE editor.

```
Map.setCenter( 8.909,33.5114, 10);
// Establish the area of interest
var AOI = ee.FeatureCollection('projects/ee-hernandezaguer3/assets/countries');

// Define spatial resolution to work with (m)
var GrainSize = 1000;
// Load Bands
// Select NDVI LANDSAT 30 meters values higher dan 0.4
var MediandwCol = ee.ImageCollection("LANDSAT/COMPOSITES/C02/T1_L2_32DAY_NDVI").filterDate('2018-01-01', '2021-12-31').select(['NDVI']).median().updateMask(ee.ImageCollection("LANDSAT/COMPOSITES/C02/T1_L2_32DAY_NDVI").filterDate('2020-01-01', '2021-12-31').select(['NDVI']).median().select('NDVI').gt(0.20));
// Load TERRACLIMATE variables
var evapo = ee.ImageCollection('IDAHO_EPSCOR/TERRACLIMATE').filter(ee.Filter.date('2018-01-01', '2021-12-31')).mean().select(['aet','tmmx','pet','pr','def','pdsi', 'soil','vpd']);
// Minimum values of NDVI LANDSAT 30 meters
var MaxNDVI2 = ee.ImageCollection("LANDSAT/COMPOSITES/C02/T1_L2_32DAY_NDVI").filterDate('2020-01-01', '2021-12-31').select('NDVI').median().rename('NDVIMedian2');
var aridity = ee.Image("projects/sat-io/open-datasets/global_ai/global_ai_yearly");
// Combine bands into a single multi-band image
var predictors = evapo.addBands(MaxNDVI2).addBands(aridity).clip(AOI);
// Select bands
var bands = ['aet','tmmx','pet','pr','def','pdsi', 'soil','vpd','b1','NDVIMedian2'];
var img = predictors.updateMask(MediandwCol.select('NDVI').gt(0)).select(bands);
// Load coordinates
var Database = ee.FeatureCollection('projects/ee-hernandezaguer3/assets/Database_global2');
Map.addLayer(Database, null, "Coordinates");
print(Database);
// create a variable called label in which we will store the property presence
var label = 'presence';
var sample = Database.randomColumn();
var validationSample = sample.filter('random > 0.8');
Map.addLayer(validationSample, null, "Coordinates validation sample");

var sample = sample.filter('random <= 0.8');
var validationData = img.select(bands).sampleRegions({
  collection: validationSample,
  properties: [label],
  scale: GrainSize,
  tileScale: 10 // Ajusta según la memoria disponible
});

// Run Random Forest Model
var classifier_disc = ee.Classifier.smileRandomForest(1000)
.train({
  features: img.select(bands).sampleRegions({
    collection: sample,
    properties: [label],
    scale: GrainSize,
    tileScale: 10 // Ajusta según la memoria disponible
  }),
  classProperty: label,
  inputProperties: bands});
print('classifier_disc',classifier_disc);
// Get information about the trained classifier.
print('Results of trained classifier', classifier_disc.explain());
// Get a confusion matrix and overall accuracy for the training sample.
var trainAccuracy = classifier_disc.confusionMatrix();
print('Training error matrix', trainAccuracy);
print('Training overall accuracy', trainAccuracy.accuracy());
// Get a confusion matrix and overall accuracy for the validation sample.
validationData = validationData.classify(classifier_disc);
var validationAccuracy = validationData.errorMatrix(label, 'classification');
print('Validation error matrix', validationAccuracy);
```

```

print('Validation accuracy', validationAccuracy.accuracy());
var classified_disc = img.select(bands).classify(classifier_disc);

var dict = classifier_disc.explain();
print('Explain 1:',dict);

var variable_importance = ee.Feature(null, ee.Dictionary(dict).get('importance'));

var chart1 =
ui.Chart.feature.byProperty(variable_importance)
.setChartType('ColumnChart')
.setOptions({
title: 'RF Variable Importance - Method 1',
legend: {position: 'none'},
hAxis: {title: 'Bands'},
vAxis: {minValue:0, title: 'Importance'},
});
print(chart1, 'Relative Importance');

var classified_disc =
img.select(bands).classify(classifier_disc).updateMask(img.select(bands).classify(classifier_disc).connectedPixelCount(200).gte(e
e.Image(1000000).divide(ee.Image.pixelArea())));
print(classified_disc.select('classification'));
var Countries = ['Algeria', 'Bahrain', 'Botswana', 'Burkina Faso', 'Cameroon',
'Cape Verde', 'Chad', 'Cyprus', 'Djibouti', 'Egypt', 'Eritrea', 'Ethiopia', 'Iraq', 'Israel',
'Jordan', 'Kenya', 'Kuwait', 'Libya', 'Mali', 'Mauritania', 'Morocco', 'Namibia', 'Niger',
'Nigeria', 'Oman', 'Palestine', 'Qatar', 'Saudi Arabia', 'Senegal', 'Somalia', 'Somaliland',
'South Africa', 'Sudan', 'Syria', 'Tunisia', 'United Arab Emirates', 'Western Sahara', 'Yemen',
'Halaib Triangle', 'Bir Tawil', 'West Bank', 'Afghanistan',
'Argentina', 'Australia', 'Cape Verde', 'Chile', 'China', 'Cyprus', 'Ecuador', 'India', 'Iran', 'Kazakhstan', 'Kyrgyzstan', 'Madagascar', 'Mexico',
'Mongolia',
'Pakistan', 'Peru', 'Russia', 'Spain', 'Turkmenistan', 'United States', 'Uzbekistan', 'Venezuela'];
var AOI = ee.FeatureCollection('USDOS/LSIB_SIMPLE/2017').filter(ee.Filter.inList('country_na', Countries));

Map.addLayer(classified_disc.updateMask(classified_disc.gt(0)),
{min: 0, max: 1, palette: ['FFFFFF','031A00']},
'Random Forest Predictions Discrete');

```