

Clasificación Supervisada

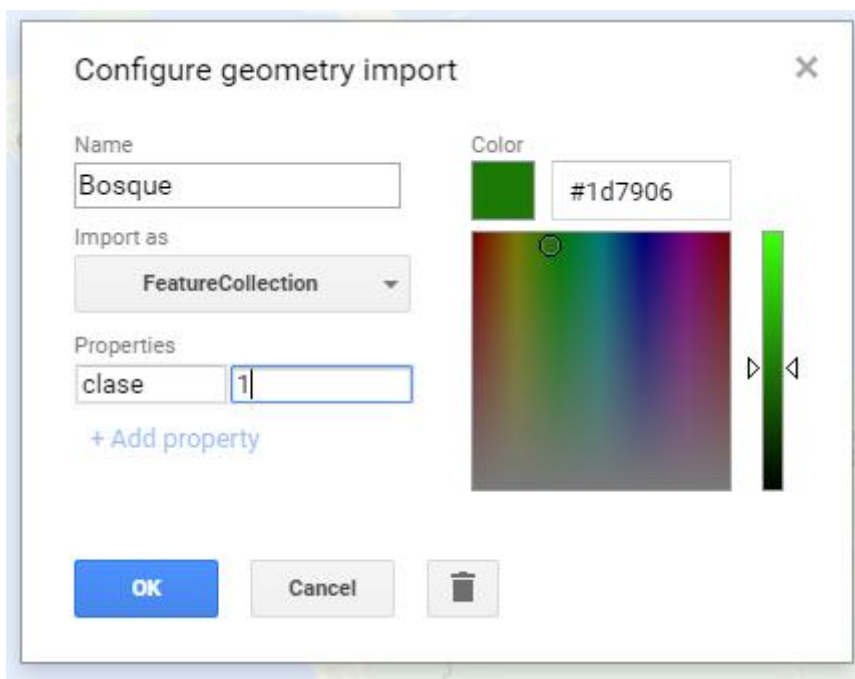
Podemos definir esta clasificación como un proceso en que pixeles de identidad conocida, ubicados dentro de las áreas de entrenamiento, se utilizan para clasificar pixeles de identidad desconocida. La clasificación supervisada involucra las siguientes etapas:

- Etapa de entrenamiento.
- Selección del algoritmo de clasificación adecuado y clasificación.
- Operaciones de post clasificación.

Todas estas etapas las podemos llevar a cabo en la plataforma GEE, siendo la última la que se realizará en menor medida, ya que para las operaciones de post clasificación finales se llevarán a cabo fuera de la misma (ordenador personal con Qgis).

Etapa de entrenamiento

Para la selección de las muestras de entrenamiento crearemos nuevas capas con la herramienta de dibujos, es importante definir que estas capas serán del tipo FeatureCollection y deben tener como un atributo de nombre "clase" y un número diferente en ese atributo, por ejemplo, bosque (clase: 1), agua(clase:2), cultivos (clase:3)...



De ser posible las muestras deben ser puntos, ya que ello facilita su manejo. Las muestras son importadas automáticamente al script con el nombre de la capa que les asignamos.

Para unir todas las muestras en una sola entidad utilizamos la función `.flatten()`

```
var muestras = ee.FeatureCollection([bosque, agua, cultivos]).flatten();
```

Selección del algoritmo de clasificación y clasificación

Una vez tengamos las muestras unidas se procede a entrenar el clasificador elegido utilizando la función `.train()`, en la plataforma GEE contamos con una gran variedad de algoritmos clasificadores entre los que podemos mencionar los algoritmos: Cart, Decision Tree, random forest, svm y otros. Para este caso haremos uso del algoritmo cart (`ee.Classifier.cart()`). Una vez se entrena el clasificador el mismo puede ser aplicado a una imagen utilizando la función `.classify()`.

A continuación se encuentra el código completo de un ejemplo:

```
//Colecciones Landsat
var Landsat5 = ee.ImageCollection('LANDSAT/LT5');
var Landsat7 = ee.ImageCollection('LANDSAT/LE7');
var Landsat8 = ee.ImageCollection('LANDSAT/LC8');

//Mosaico Fecha 1
var LandsatFecha1=Landsat8.filterBounds(geometry)
                           .filterDate('2016-01-01', '2016-12-31')
                           .filter(ee.Filter.calendarRange(8,3,"month"))
                           .filterMetadata('CLOUD_COVER','less_than', 40);

var mosaicoSLC1 = ee.Algorithms.Landsat.simpleComposite(LandsatFecha1, 60, 10, 40, true);

var mosaicoCorte1 = mosaicoSLC1.clip(geometry);

Map.addLayer (mosaicoCorte1, {bands: ['B5','B4','B3'], min: [0,0,0], max: [0.5, 0.3, 0.3] },
'Mosaico L8 2016', true );

//Clasificación Supervisada
var muestras = ee.FeatureCollection([Bosque, NoBosque]).flatten() ;
var training = mosaicoCorte1.sampleRegions({
  collection: muestras,
  properties: ['clase'],
  scale: 30
});

var bandas = ['B2','B3','B4','B5','B6','B7'];
var trained = ee.Classifier.cart().train(training, 'clase', bandas);
var classified = mosaicoCorte1.select(bandas).classify(trained).clip(geometry);
Map.addLayer(classified, {min:1, max:2, palette: ['339820', 'e6f0c2']}, 'Clasificación');
```

O podemos acceder a un ejemplo con muestras siguiendo el siguiente enlace

<https://code.earthengine.google.com/03911b0b36f87b574aad46ed700ee7cb>

Utilización de Mascaras

Para mejorar la clasificación se pueden utilizar productos de teledetección como los que encontramos en la plataforma. En este caso utilizaremos unas capas de cobertura de agua global (**JRC Global Surface Water Mapping Layers v1.0 (1984-2015)**) y cobertura de áreas urbanas globales (**Nombre Cobertura**) para enmascarar dichas categorías en la clasificación obtenida anteriormente.

```
//Clasificación Supervisada
var muestras = ee.FeatureCollection([Bosque, NoBosque]).flatten() ;
var training = mosaicoCorte1.sampleRegions({
  collection: muestras,
  properties: ['clase'],
  scale: 30
});

var bandas = ['B2','B3','B4','B5','B6','B7'];
var trained = ee.Classifier.cart().train(training, 'clase', bandas);
var classified = mosaicoCorte1.select(bandas).classify(trained).clip(geometry);
Map.addLayer(classified, {min:1, max:2, palette: ['339820', 'e6f0c2']}, 'Clasificación');

//Mascaras Ciudades y Agua
var ciudades = ee.Image('users/christinacorbane/MT_comp_proj').clip(geometry); //1 Agua 2 Otros
3456 Ciudades
var ciudades2 = ciudades.gte(3);
var water = ee.Image('JRC/GSW1_0/GlobalSurfaceWater').clip(geometry);
var water2 = water.select('transition').lte(2).or(water.select('transition').eq(4))
.or(water.select('transition').eq(5)).or(water.select('transition').eq(7))
.or(water.select('transition').eq(8));

//Aplicación de Mascaras
var classified_correct = classified.where(water2.eq(1),3);
var classified_correct2 = classified_correct.where(ciudades2.eq(1),4);
Map.addLayer(classified_correct2,{ min:1, max:4, palette: ['339820','e6f0c2','0446ff','d63000']},
'Clasificación con Mascaras');
```

O podemos acceder a un ejemplo siguiendo el siguiente enlace

<https://code.earthengine.google.com/a039ef75f4047d6d426f05157190e141>

Como se puede observar en estas líneas se introdujeron varias funciones nuevas cuya función será detallada a continuación:

Las funciones `eq()`, `neq()`, `gt()`, `gte()`, `lt()`, `lte()` corresponden a igual, no igual, mayor, mayor o igual, menor y menor o igual, respectivamente, y son utilizadas para seleccionar pixeles con determinados valores dentro de una capa raster.

Por ejemplo para el caso de la capa de áreas de urbanas los valores de pixeles son los siguientes:

1 – Agua, 2 – Otros, 3 – Ciudades, 4 – Ciudades, 5 – Ciudades, 6 – Ciudades

Por lo tanto para crear una capa en la que estén seleccionados solo los valores correspondientes a áreas urbanas tendríamos que seleccionar los valores 3, 4,5 y 6. Lo cual se puede hacer de varias formas. Obteniendo con todas ellas el mismo resultado.

```
//Mascara Ciudades
var ciudades = ee.Image('users/christinacorbane/MT_comp_proj').clip(geometry);
var ciudades1 = ciudades.gt(2);
var ciudades2 = ciudades.gte(3);
var ciudades3 = ciudades.eq(3).or(ciudades.eq(4)).or(ciudades.eq(5)).or(ciudades.eq(6));

Map.addLayer(ciudades1.randomVisualizer(), {}, 'Ciudades Mayor a');
Map.addLayer(ciudades2.randomVisualizer(), {}, 'Ciudades Mayor o igual a');
Map.addLayer(ciudades3.randomVisualizer(), {}, 'Ciudades Igual a');
```

Como se puede observar en el último ejemplo se introdujo además la función `.or()` la cual indica que serán seleccionados los pixeles que cumplan con **alguna** de las condiciones indicadas.

Para la creación de la capa de aguas superficiales se utilizó la capa **JRC Global Surface Water Mapping Layers v1.0 (1984-2015)** con ID 'JRC/GSW1_0/GlobalSurfaceWater', la misma es una imagen que cuenta con 7 bandas, tanto estas bandas como sus valores pueden observarse en la sección de información de la capa en la plataforma GEE

https://code.earthengine.google.com/dataset/JRC/GSW1_0/GlobalSurfaceWater

Para este ejemplo en particular solo se utilizó la banda correspondiente a 'transition' utilizando los valores 0,1,2,4,5,7 y 8. En la imagen a continuación se detallan las categorías correspondiente a dicho valores.

transition	Categorical classification of change between first and last year.	0 - No change 1 - Permanent 2 - New permanent 3 - Lost permanent 4 - Seasonal 5 - New seasonal 6 - Lost seasonal 7 - Seasonal to permanent 8 - Permanent to seasonal 9 - Ephemeral permanent 10 - Ephemeral seasonal
------------	-------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Para la selección de dichos valores se pueden utilizar varias sentencias, las cuales se pueden observar a continuación:

```
//Mascara Agua
var agua = ee.Image('JRC/GSW1_0/GlobalSurfaceWater').clip(geometry);
var agua1 = agua.select('transition').lte(2).or(agua.select('transition').eq(4))
.or(agua.select('transition').eq(5)).or(agua.select('transition').eq(7))
.or(agua.select('transition').eq(8));
var agua2 = agua.select('transition').lte(8).and(agua.select('transition').neq(3))
.and(agua.select('transition').neq(6));

Map.addLayer(agua1.randomVisualizer(), {}, 'Agua igual a');
Map.addLayer(agua2.randomVisualizer(), {}, 'Agua no igual a');
```

Como se puede observar en el último ejemplo se introdujo además la función `.and()` la cual indica que sólo serán seleccionados los píxeles que cumplan con **todas** de las condiciones indicadas.

Por último, la función `.where()` es una función que permite reemplazar los píxeles dentro de un archivo raster basándose en una determinada condición. Los argumentos de esta condición son

raster.`.where`(condición, valor a asignar)

```
//Aplicación de Mascaras
var classified_correct = classified.where(water2.eq(1),3);
var classified_correct2 = classified_correct.where(ciudades2.eq(1),4);
Map.addLayer(classified_correct2,{ min:1, max:4, palette: ['339820','e6f0c2','0446ff','d63000']},
'Clasificación con Mascaras');
```

En el ejemplo anterior se reemplazaron los píxeles en la imagen clasificada (**classified**) donde la capa de agua (**water2**) tuviera un valor igual a 1, el nuevo valor asignado a dichos píxeles es de **3**.

Así también se reemplazaron los píxeles en la imagen clasificada corregida (**classified_correct**) donde la capa de áreas urbanas (**ciudades2**) tuviera un valor igual a 1, el nuevo valor asignado a dichos píxeles es de **4**.

Autor

José Mario Serafini Gauto

Jose.SerafiniGauto@fao.org

jose.serafinig@gmail.com

Contactos auxiliares

Adolfo Kindgard

adolfo.kindgard@fao.org

kindgard@gmail.com

Erith Muñoz

Erith.MunozRios@fao.org

erith7@gmail.com