



EXERCISE 3

Running a random forest model in Earth Engine

Introduction

The Earth Engine Code Editor comes fully loaded with a diverse collection of tools for processing and analyzing spatial data. These tools are accessible as JavaScript functions and methods. In addition to GEE functions specific for the analysis of spatial data, most standard JavaScript functions are also included. The availability of generic and specific functions allows users to create customized analysis routines and develop highly sophisticated customized scripts. In this exercise, we will learn about a routine developed for building and running a random forests model.

The following exercise will walk you through the process of creating a simple classification that uses the same NAIP imagery from eastern Oregon as was used in the previous exercise, however the data will be accessed and analyzed in Earth Engine's cloud-computing environment instead of locally.

Objectives

- Orient yourself to the Earth Engine script
- Execute a random forest classification in Earth Engine
- Export output and compare to the RStudio classification

Required Data

- none

Prerequisites

- You have ESRI ArcGIS 10.2 installed on your computer and have basic understanding of how to use the software
- You have completed the previous exercise
- You have a Google Drive account set up (<https://drive.google.com>)
- You have a Google Earth developer account (<https://signup.earthengine.google.com/#/>)





Table of Contents

Part 1: Investigate the random forests script	3
Part 2: Prepare a classification for the study area	4
Part 3: Download the classification	6
Part 4: Compare classifications	7



Part 1: Investigate the random forests script

A. Open the script link

1. Use the link below to run a prepared script
2. <https://code.earthengine.google.com/a39ab832aa7872d9a340234f1391cf61>

Note: you may need to grant Earth Engine permission to write to your Google Drive account and view and manage your Earth Engine data. If you receive this message, click Allow.

3. Click **Show Code** and review the script.

Note: the user editable variables are peppered throughout the script and are marked with comments (in Earth Engine, comments are marked with two forward slashes (//)).

B. Explore the following:

1. **Line 3** – centers the script on geographic coordinates
2. **Line 7** – creates a variable, **naip**, using the geographic coordinates specified and a date filter set by the user
3. **Line 16** – creates a variable, **points**, which extracts training points for the classification from an asset
4. **Line 19** – creates a variable, **training**, which intersects points with the NAIP imagery
5. **Line 22** – builds the random forests model

Note: Line 22 contains the line of code to specify and train (line 23) the classifier. The random forest classifier is called using `ee.Classifier.randomForest(numberOfTrees, variablesPerSplit, minLeafPopulation, bagFraction, outOfBagMode, seed)`. In our script, we only adjust two arguments: `numberOfTrees` and `outOfBagMode`. We leave all others as the default.

```
ee.Classifier.randomForest(numberOfTrees, variablesPerSplit, minLeafPopulation, bagFraction, outOfBagMode, seed) x
```

Creates an empty Rifle Serial classifier, which uses the Random Forest algorithm.

Arguments:

- **numberOfTrees** (Integer, default: 1):
The number of Rifle decision trees to create per class.
- **variablesPerSplit** (Integer, default: 0):
The number of variables per split. If set to 0 (default), defaults to the square root of the number of variables.
- **minLeafPopulation** (Integer, default: 1):
The minimum size of a terminal node.
- **bagFraction** (Float, default: 0.5):
The fraction of input to bag per tree.
- **outOfBagMode** (Boolean, default: false):
Whether the classifier should run in out-of-bag mode.
- **seed** (Integer, default: 0):
Random seed.

Returns: Classifier

6. **Line 26** – applies the random forests model to the imagery
7. Review the rest of the script on your own.

Part 2: Prepare a classification for the study area

There are four main parameters that the user must input in order to successfully run this script: (1) geographic coordinates of general area to be mapped; (2) filter dates with which to search for relevant imagery; (3) an asset containing the points to be used for training the model; and (4) the field/attribute name from the asset that the classifier should be trained on.

A. Review editable variables

1. Set up the user editable variables according to the code below:
 - i. The point location of the **roi** variable should be set as **-118.9819, 43.9054** – this directs the script to 43.9054°N, 118.9819°W. This variable is found on **line 3**, and these coordinates are not strings, so no quotes are required.

Note: Our study area falls within UTM Zone 11N, which is also referenced by the EPSG code 32611. If you edit this script for another project, and your data falls within a different UTM Zone, you should look up the appropriate EPSG code at <http://www.spatialreference.org/> and adjust the variable on line 4 accordingly.

- ii. The NAIP imagery that we're using is from 2012 – the **filter start date** should be set to **2012-01-01** and the **filter end date** should be set to **2012-12-31**. These parameters are found on **line 9**, with the end date following the start date. The dates should be strings (i.e., within single quotes).
- iii. Finally, the user must specify an asset as an Import, which is essentially just a shapefile that has been uploaded. In this case, the shapefile converted to Earth Engine asset, called Wenatchee_Training, contains a field or attribute called Class_ID. This column represents unique identifiers for each category in the classification (1-4 representing ground, shadow, shrub, and tree).
- iv. If you have a different asset you would like to use, you can import it into the script and reference it on line 16.
- v. On **lines 19** and **23** of the script, we must insert the field/attribute name (e.g., **Class_ID**) from the asset that we wish to map. These lines will extract the spectral values at the training points and train the model based on these data, respectively. Place the field/attribute line where specified in the script.

Note: the number (e.g., 1) following **Class_ID** in **line 19** refers to the sampling resolution. Since NAIP imagery has a spatial resolution of 1 meter, this parameter is set to 1. If we were working with Landsat imagery, this parameter would be set to 30. If you wish to adjust the total number of trees used to create the model, adjust the integer found on line 22.

2. Your script should now look like the screen capture shown below.

```

Imports (1 entry)
var wenatchee_training: Table users/brennaschwert/Courses/RF/Wenatchee_Training

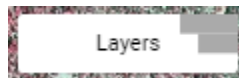
1 // Define a region of interest as a point. Change the coordinates
2 // to get a classification of any place where there is imagery.
3 var roi = ee.Geometry.Point(-118.9819, 43.9054);
4 var crs = 'EPSG:32611';
5
6 // Load NAIP input imagery.
7 var naip = ee.Image(ee.ImageCollection('USDA/NAIP/DOQQ')
8 // Filter to get only one year of images.
9 .filterDate('2012-01-01', '2012-12-31')
10 // Filter to get only images under the region of interest.
11 .filterBounds(roi)
12 // Get the first scene.
13 .first());
14
15 //Load training points. The property "Class_Num" stores the known labels
16 var points = ee.FeatureCollection(wenatchee_training, 'geometry');
17
18 //Intersect the points with the imagery to get training data
19 var training = naip.sampleRegions(points, ['Class_ID'], 1);
20
21 // Make a Random Forest classifier and train it.
22 var classifier = ee.Classifier.randomForest(1000, 0, 1, 0.5, true, 0)
23 | .train(training, 'Class_ID');
24

```

- Review the rest of the script. There may be some other parameters that you may need or want to adjust, such as the palette used to display the classification (found starting on line 34) or export parameters (found starting on line 55).

B. Run the script

- Run the script by pressing the **Run** button.
- Wait for the layers to load in the map window. The bars will be partially gray while loading/computing, as shown below.



- Hover over **Layers** and check/uncheck the boxes to view/hide layers and examine results. Make any adjustments to the script that you deem necessary and rerun.
- Click on the **Tasks** tab and review the two rasters available for export:
 - NAIP_GEE_predict1000** – Google Earth Engine’s random forests prediction of the NAIP imagery using 1000 trees.
 - NAIP_subset** – Subset of the NAIP imagery to match the imagery used in the previous exercise.
- Click **Run** next to either or both layers to write the data to your Google Drive account. Remember that you already have a local copy of the NAIP imagery, so you might not want to export that again.
 - Adjust the filename (if desired). This will be the name of the file as it appears in your Google Drive.
 - Leave the **Resolution** as the default.
 - Click **Run**.

Task: Initiate image export ×

Task name (no spaces)
NAIP_subset

Resolution
Scale (m/px) ▼ 1

Drive folder
my-drive-folder-name or blank for root

Filename
NAIP_subset

Run Cancel

Note: This export process may take a considerable amount of time and is largely dependent on the size of the data that you're exporting.

- While the export is running, you will see a **spinning gear icon** next to the layer name in the **Tasks** tab. When finished, the spinning gear will turn into a **checkmark**. In the image shown below, the classification is being exported (gear icon shown), and the NAIP imagery has yet to be exported (Run button shown).



Part 3: Download the classification

You've exported your data from Google Earth Engine, but it's still in the cloud! In order to access the data locally, we need to download it from Google Drive.

A. Download the classification

- Go to your Google Drive account: <https://drive.google.com>
- NAIP_GEE_predict1000.tif** should be at the top of your files list, and the file size should be approximately 279 KB – if your raster seems smaller than this, rerun the export.

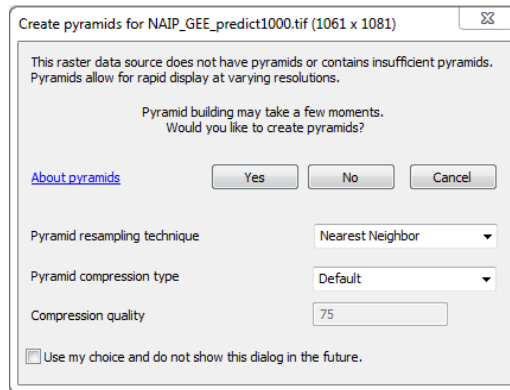


- Right-click on the file **NAIP_GEE_predict1000.tif** and select **Download**. This will send a copy of the raster from the cloud to your local Downloads folder.
- After the download is complete, you may need to free up some space in your Google Drive. Right-click on the file and select **Remove**. This will move the file to your **Trash**.
- To permanently delete the file and actually free up that space in your Google Drive, click on **Trash**, right-click on the file, and select **Delete Forever**. You will receive a warning message stating that you cannot undo this action, select **Delete Forever**.

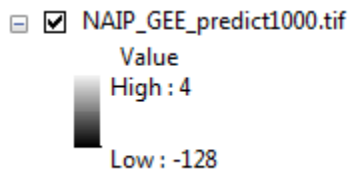
Part 4: Compare classifications

A. Load classification in ArcMap and compare

1. When the download is complete, open the file in ArcMap. You may be prompted to build pyramids for the file – choose **Yes**.



2. By default, the raster will display with a continuous symbology – we need to adjust it to a categorical display.



- i. Double-click on the file **NAIP_GEE_predict1000.tif** in the **Table of Contents** to open the **Layer Properties** window.
- ii. Select the **Symbology** tab, and click **Unique Values** under the **Show** menu on the left.
- iii. You will receive a message asking to build a raster attribute table – select **Yes**.
- iv. You will now see five unique values. Apply the following **labels** to each:
 - (a) -128: No Data
 - (b) 1: Ground
 - (c) 2: Shadow
 - (d) 3: Shrub
 - (e) 4: Tree
- v. Double-click on the rectangles of color representing each class's **symbol** to change the color. Make the colors match those that were used to display the classification from RStudio:
 - (a) No Data: No color
 - (b) Ground: Light gray
 - (c) Shadow: Black

- (d) Shrub: Brown
- (e) Tree: Green
- vi. Click **OK** to accept Symbology changes.



3. If you no longer have the RStudio classification (without NDVI) open, load the file (**predict_NAIP_subset.tif**) into ArcMap and adjust the symbology as necessary such that it matches the Google Earth Engine output.
4. In the **Effects** toolbar, make sure that your top layer (**predict_NAIP_subset.tif**) is selected from the drop-down menu.
5. Zoom in to an area of interest, select the **Swipe** tool from the **Effects** toolbar, and slide the cursor between the two outputs. How do they compare?

Note: The algorithm for random forests is proprietary and trademarked, so it is not available for distribution in software packages like R/RStudio or within Google Earth Engine. However, the algorithm is well-outlined and freely available, so there are many versions that are negligibly different. You should not notice too many differences between the classifications – any differences you do see are in small part because of the use of slightly different algorithms, but mostly because of the randomization involved with tree generation, which will always yield slightly different results (even from the same algorithm).

Congratulations! You have completed this exercise, and you now know how to create, export, and download classifications from Google Earth Engine using a random forests classifier.