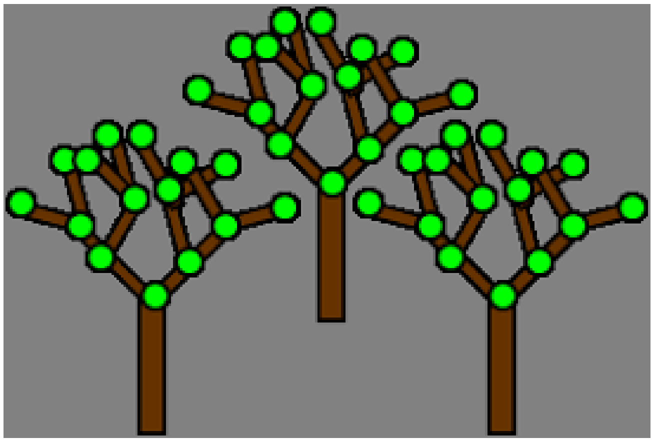
EXERCISE 5
Run Your Own Random Forest Regression & Classification



Introduction

The previous exercises have taught us how to 1) access Landsat Imagery to produce a seasonal composite, 2) calculate spectral indices using Landsat spectral bands, and 3) use other GEE resources to compile topographic and climatic data. During exercise 4 we used that information to run a Random Forest classification and regression, so now you're all ready to input your own data and run your own model.

First, assess your data and decide if you should be running a **classification** or a **regression**. Are you trying to identify discrete classes (soil type, etc.)? If so, you should be running a **classification**. Are you hoping to model a continuous variable (soil pH, redox depth, soil moisture, etc.)? If yes, then you should be running a **regression**.

Objectives

- Use NRCS_DSM library of functions to load in Landsat composite and stack predictor layers
- Modify exercise 4.1 and/or 4.2 to run a Random Forest classification and/or regression on your own datasets and ROIs
    - Conduct model accuracy assessment by interpreting useful statistics and figures
    - Display final classification and legend on map

Required Data:

- Your own boundary shapefile for AOI
- Your own pedons shapefile (point training data)
  OR
- **GNP_boundary.shp** – shapefile representing example area of interest
- **GNP_pedons.shp** – shapefile of training data for Glacier National Park, MT

Prerequisites

- **Completion of Exercise 1-4 (you can review code by accessing completed scripts in the course repository)**
- **Google Chrome installed on your machine**
- **An approved Google Earth Engine account**
- **Follow the links below to gain read access to the GEE code repositories we will refer to in the script.**
  - Click here to gain access to the GTAC module repository
  - Click here to gain access to the GTAC training repository

USDA Non-Discrimination Statement

In accordance with Federal civil rights law and U.S. Department of Agriculture (USDA) civil rights regulations and policies, the USDA, its Agencies, offices, and employees, and institutions participating in or administering USDA programs are prohibited from discriminating based on race, color, national origin, religion, sex, gender identity (including gender expression), sexual orientation, disability, age, marital status, family/parental status, income derived from a public assistance program, political beliefs, or reprisal or retaliation for prior civil rights activity, in any program or activity conducted or funded by USDA (not all bases apply to all programs). Remedies and complaint filing deadlines vary by program or incident.

Persons with disabilities who require alternative means of communication for program information (e.g., Braille, large print, audiotape, American Sign Language, etc.) should contact the responsible Agency or USDA's TARGET Center at (202) 720-2600 (voice and TTY) or contact USDA through the Federal Relay Service at (800) 877-8339. Additionally, program information may be made available in languages other than English.

To file a program discrimination complaint, complete the USDA Program Discrimination Complaint Form, AD-3027, found online at How to File a Program Discrimination Complaint and at any USDA office or write a letter addressed to USDA and provide in the letter all of the information requested in the form. To request a copy of the complaint form, call (866) 632-9992. Submit your completed form or letter to USDA by: (1) mail: U.S. Department of Agriculture, Office of the Assistant Secretary for Civil Rights, 1400 Independence Avenue, SW, Washington, D.C. 20250-9410; (2) fax: (202) 690-7442; or (3) email: program.intake@usda.gov.

USDA is an equal opportunity provider, employer, and lender.

Table of Contents

Part 1: Run your own Random Forest Classification

A. Load & re-save your script

1. For this exercise, we will be re-using the scripts that we assembled during exercise 4. We don't want to change them permanently though, so copy and paste the code from exercise 4.1 into a new file and re-save it. Now that we have a copy, we can change it without worry.
2. If for some reason you don't have your own script from exercise 4.1, there's a completed script in the (enter correct file location here) that you can easily copy into a new file.

B. Load in your own data as assets

1. To run this classification smoothly, you will need to upload two of your own shapefiles as assets – a **boundary shapefile** around your ROI, and a **pedons shapefile** that has points of training data to use. If you don't have your own dataset, you can use the GNP_MT dataset provided.
2. Before we load our data into GEE, we need to do some data management. Since we're running a classification, the goal probably is to classify soil types or the like. If the property of interest is **NOT** a number, we must **CREATE** a new numeric field that directly corresponds to the soil property we're hoping to classify. Remember how we used the 'Class_num' field to classify our image in exercise 4.1? You must ensure you have this field to successfully run the classification. To make things easier, we've put together a Python script that adds the "Class_num" field to your dataset.
   i. Navigate to the course downloads folder, find the file called **classFieldToNumerical.py**, right-click, choose **Save link as**, as save it to your local machine. Once downloaded, find the file, right click on it, and select "**Edit with IDLE**" or "**Edit with IDLE (ArcGIS Pro)**".
   ii. Once the python script is open, scroll to the bottom where it says "**EDIT PARAMETERS BELOW**", change the first two parameters: the first one being the *path to the pedons dataset shapefile*, and the second being the *name of the attribute of interest* (soil type, etc).

NOTE: technically there is a third editable parameter – the new field name, which is set to "Class_num" by default – leave this as is.

   iii. Once you edited the two parameters, click "**Run"** and then **"Run Module**". The code should run and you should have another field labeled "Class_num" in your pedons dataset.
3. Once you've done this, load your boundary and pedons shapefiles into GEE as assets like we did in previous activities, then import them into your script. Make sure you rename your boundary and table features once they're imported to something similar to what we used during exercise 4.1.
   i. Make sure you're familiar with your pedons dataset. You can inspect it by clicking the file name in the **Assets** tab and navigating to the **FEATURES** tab.

C. Edit your script

1. Now that we have our ROI boundary and pedon datset loaded in, we can start adapting the rest of the script.
2. First, we must change the '**User Editable Variable'** in Part 1.
   i. Rename the definition of the variable **compositeArea** so that it reflects the new name of your ROI boundary file. Example – *filename*.geometry().boundary()
   ii. Rename the **roiName** to match your new ROI
   iii. Change the **crs** variable depending on your ROI. Visit SpatialReference.org to search for the appropriate projection/ESPG number for your study region.
3. In Part 2 we load in and organize the training data. In the first line of code, change '**TX_pedons**' to whatever your new pedon imported file is called:

```
var data = ee.FeatureCollection(TX_pedons, 'geometry');
```

   i. Depending on how many training points you have, it's safe to keep the 70%/30% split of points for training/validation. If you happen to have a validation dataset, you could import it for the validation (in which case you would not split your pedons dataset – for reference, exercise 4.2 loads in the pedons dataset for 100% training).
4. You can leave Part 3 as it is.
5. In Part 4, we must edit the palette so that it aligns with the class being identified.
   i. In exercise 4.1, we had seven soil classes – each class name corresponding with a number – that had a color assigned to it. Make sure you comment your code so that you know which color belongs to which class! See code below:

```
var palette = [
  '8B0000', // Barlite 1
  'FFA500', // Berrend 2
  '4169E1', // Chilimol 3
  '006400', // Marfa 4
  '808000', // Murray 5
  'BC8F8F', // Musquiz 6
  'FFFF00', // Phantom 7
];
```

   ii. Depending on the amount and names of your soil classes, your palettes may have more or less values. Visit Wikipedia's page on Hex Colors to see available color options.
   iii. In the line of code below, remember to change the palette **min** and **max** values. In our case, we had 7 soil classes – so our min is 1, and max is 7.

```
Map.addLayer(classified, {palette: palette, min: 1, max: 7}, 'classification');
```

   iv. Find the following chunk of code located towards the end of part 4:

```
 // Display names for each legend item updated here
```

```
legend.add(makeRow('8B0000', 'Barlite (1)')); //dark red
legend.add(makeRow('FFA500', 'Berrend (2)')); //orange
legend.add(makeRow('4169E1', 'Chilimol (3)')); //blue
legend.add(makeRow('006400', 'Marfa (4)')); //green
legend.add(makeRow('808000', 'Murray (5)')); //olive
legend.add(makeRow('BC8F8F', 'Musquiz (6)')); //rose brown
legend.add(makeRow('FFFF00', 'Phantom (7)')); //yellow
```

(a) Edit the legend so it matches the changes you made to the palette.

6. In Part 5 of the script, we create accuracy assessment stats in figures, then perform a validation – there is nothing to edit in either of these sections.

D. Great job -- Save your script, and run it!

Part 2: Run your own Random Forest Regression

A. Load & re-save your script

1. For this section, we will also be re-using the scripts that we assembled during exercise 4. We don't want to change them permanently though, so copy and paste the code from exercise 4.2 into a new file and re-save it.
2. If for some reason you don't have your own script from exercise 4.2, there's a completed script in the course repository that you can easily copy into a new file.

B. Load in your own data as assets

1. To run this regression smoothly, you will need to upload two of your own shapefiles as assets – a **boundary shapefile** around your ROI, and a **pedons shapefile** that has points of training data to use. If you don't have your own dataset, you can use the GNP_MT dataset provided.
2. Load these datasets into GEE as assets like we did in previous activities, and import them into your script. Make sure you rename your boundary and table features once they're imported to something similar to what we used during exercise 4.2.
3. Since we're running a regression, you must be modelling a continuous variable (soil pH, soil moisture, etc). Make sure you're familiar with your pedons dataset, and you've decided what variable you plan to model. You can inspect it by clicking the file name in the **Assets** tab and navigating to the **FEATURES** tab.

C. Change the class property of interest

1. Now we must change the class property of interest throughout our script. In exercise 4.2, we were modelling the 'REDOX_CM' property. To find all the lines you need to change, press Ctrl+F and type 'REDOX_CM', then replace each one with your own soil property of interest.

***BONUS:** If you want make this even easier and develop your coding skill a bit more, define a new variable at the beginning of your script so you don't have to hunt for each property input throughout your script.*

D. Great job -- Save your script, and run it!

Congratulations! You have successfully completed this exercise. You have used a variety of techniques to perform random forest classification and/or regression in Google Earth Engine.