

AI Course

Capstone Project Final Code

For students (instructor review required)

Group Members:

Muhammad Hussnain

Ghayur Abbas

Muhammad Shoaib

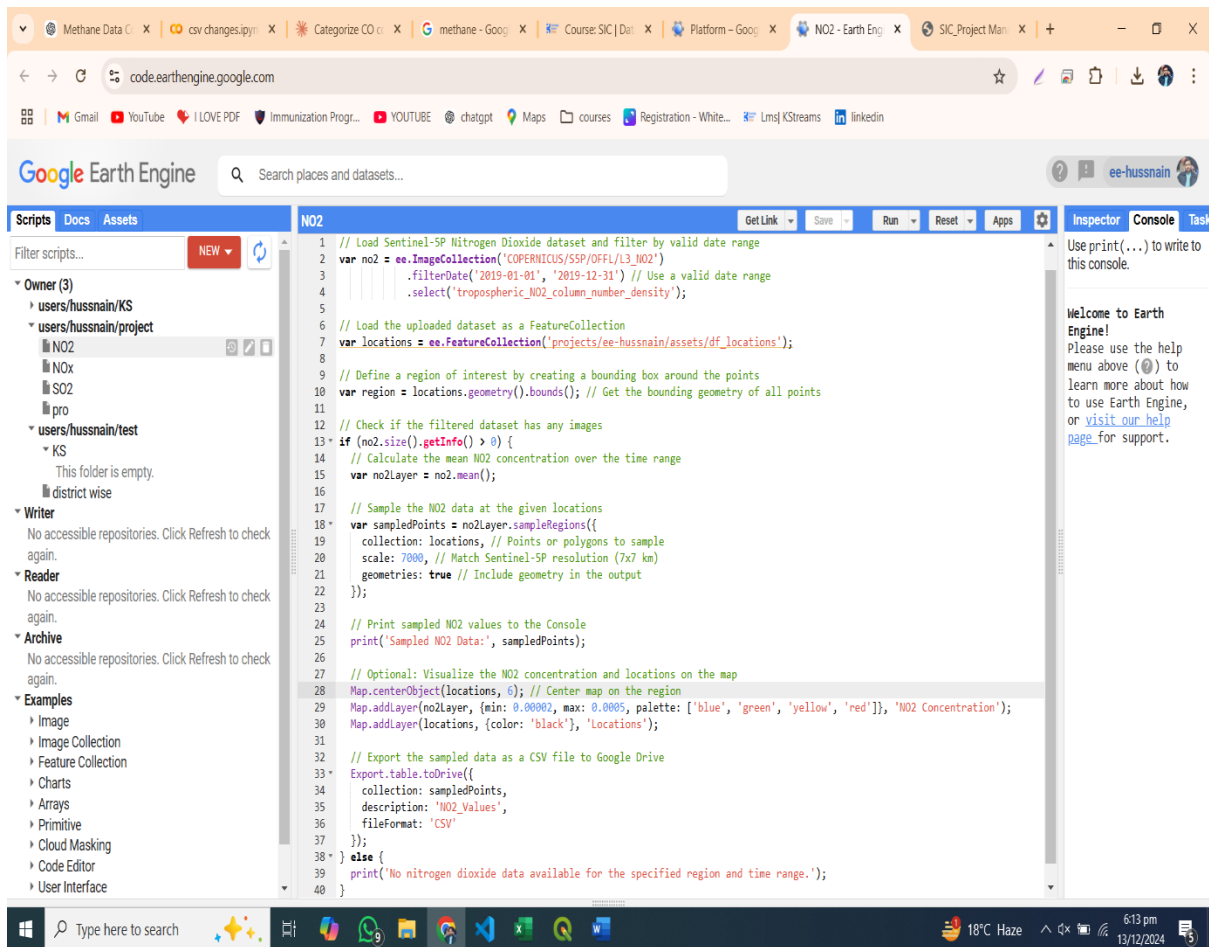
©2023 SAMSUNG. All rights reserved.

Samsung Electronics Corporate Citizenship Office holds the copyright of this document.

This document is a literary property protected by copyright law so reprint and reproduction without permission are prohibited.

To use this document other than the curriculum of Samsung Innovation Campus, you must receive written consent from copyright holder.

NO2 Gas is Detected by Google Earth Engine.



Code of NO2 Gas:

```
// Load Sentinel-5P Nitrogen Dioxide dataset and filter by valid date range
var no2 = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_NO2')
    .filterDate('2019-01-01', '2019-12-31') // Use a valid date range
    .select('tropospheric_NO2_column_number_density');

// Load the uploaded dataset as a FeatureCollection
var locations = ee.FeatureCollection('projects/ee-hussnain/assets/df_locations');

// Define a region of interest by creating a bounding box around the points
var region = locations.geometry().bounds(); // Get the bounding geometry of all
points

// Check if the filtered dataset has any images
if (no2.size().getInfo() > 0) {
    // Calculate the mean NO2 concentration over the time range
    var no2Layer = no2.mean();

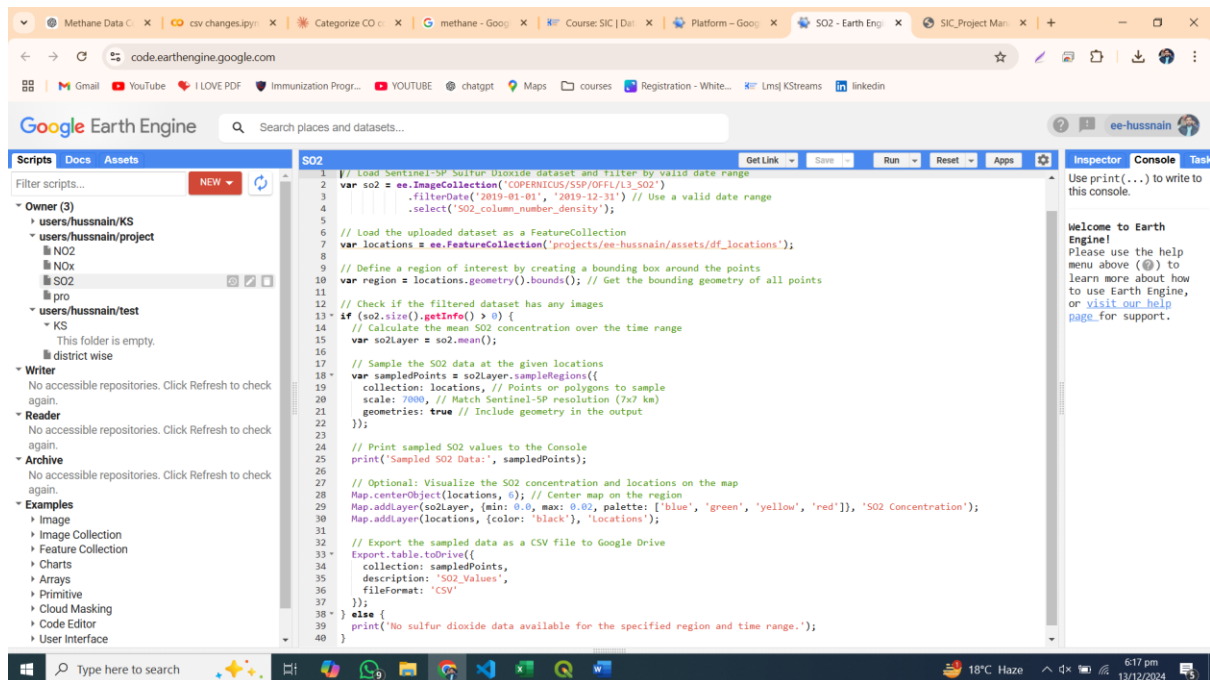
    // Sample the NO2 data at the given locations
    var sampledPoints = no2Layer.sampleRegions({
        collection: locations, // Points or polygons to sample
        scale: 7000, // Match Sentinel-5P resolution (7x7 km)
        geometries: true // Include geometry in the output
    });

    // Print sampled NO2 values to the Console
    print('Sampled NO2 Data:', sampledPoints);

    // Optional: Visualize the NO2 concentration and locations on the map
    Map.centerObject(locations, 6); // Center map on the region
    Map.addLayer(no2Layer, {min: 0.00002, max: 0.0005, palette: ['blue', 'green',
'yellow', 'red']}, 'NO2 Concentration');
    Map.addLayer(locations, {color: 'black'}, 'Locations');

    // Export the sampled data as a CSV file to Google Drive
    Export.table.toDrive({
        collection: sampledPoints,
        description: 'NO2_Values',
        fileFormat: 'CSV'
    });
} else {
    print('No nitrogen dioxide data available for the specified region and time
range.');
```

SO2 Gas is Detected by Google Earth Engine.



CODE:

```
// Load Sentinel-5P Sulfur Dioxide dataset and filter by valid date range
var so2 = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_SO2')
    .filterDate('2019-01-01', '2019-12-31') // Use a valid date range
    .select('SO2_column_number_density');

// Load the uploaded dataset as a FeatureCollection
var locations = ee.FeatureCollection('projects/ee-hussnain/assets/df_locations');

// Define a region of interest by creating a bounding box around the points
var region = locations.geometry().bounds(); // Get the bounding geometry of all points

// Check if the filtered dataset has any images
if (so2.size().getInfo() > 0) {
    // Calculate the mean SO2 concentration over the time range
    var so2Layer = so2.mean();

    // Sample the SO2 data at the given locations
    var sampledPoints = so2Layer.sampleRegions({
        collection: locations, // Points or polygons to sample
        scale: 7000, // Match Sentinel-5P resolution (7x7 km)
        geometries: true // Include geometry in the output
    });

    // Print sampled SO2 values to the Console
    print('Sampled SO2 Data:', sampledPoints);

    // Optional: Visualize the SO2 concentration and locations on the map
    Map.centerObject(locations, 6); // Center map on the region
    Map.addLayer(so2Layer, {min: 0.0, max: 0.02, palette: ['blue', 'green', 'yellow', 'red']}, 'SO2 Concentration');
    Map.addLayer(locations, {color: 'black'}, 'Locations');

    // Export the sampled data as a CSV file to Google Drive
    Export.table.toDrive({
        collection: sampledPoints,
        description: 'SO2_Values',
    });
}
```

```
        fileFormat: 'CSV'
    });
} else {
    print('No sulfur dioxide data available for the specified region and time
range.');
```

CO Gas is Detected by Google Earth Engine.

CODE:

```
// Load Sentinel-5P Carbon Monoxide dataset and filter by valid date
range
var carbonMonoxide = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_CO')
    .filterDate('2019-01-01', '2019-12-31') //
Use a valid date range
    .select('CO_column_number_density');

// Load the uploaded dataset as a FeatureCollection
var locations = ee.FeatureCollection('projects/ee-
ghayurabbas13/assets/df2');

// Define a region of interest by creating a bounding box around the
points
var region = locations.geometry().bounds(); // Get the bounding
geometry of all points

// Check if the filtered dataset has any images
if (carbonMonoxide.size().getInfo() > 0) {
    // Calculate the mean CO concentration over the time range
    var coLayer = carbonMonoxide.mean();

    // Sample the CO data at the given locations
    var sampledPoints = coLayer.sampleRegions({
        collection: locations, // Points or polygons to sample
        scale: 7000, // Match Sentinel-5P resolution (7x7 km)
        geometries: true // Include geometry in the output
    });

    // Print sampled CO values to the Console
    print('Sampled CO Data:', sampledPoints);

    // Optional: Visualize the CO concentration and locations on the
map
    Map.centerObject(locations, 6); // Center map on the region
    Map.addLayer(coLayer, {min: 0.03, max: 0.05, palette: ['blue',
'green', 'yellow', 'red']}, 'CO Concentration');
    Map.addLayer(locations, {color: 'black'}, 'Locations');

    // Export the sampled data as a CSV file to Google Drive
    Export.table.toDrive({
        collection: sampledPoints,
```

```

        description: 'Sampled_CO_Values',
        fileFormat: 'CSV'
    });
} else {
    print('No carbon monoxide data available for the specified region
and time range.');
```

CH4 Gas is Detected by Google Earth Engine.

The screenshot displays the Google Earth Engine web interface. The left sidebar shows the 'Scripts' tab with a list of scripts under the 'pro' folder. The main area shows a script titled 'pro' with the following code:

```

1 // Load Sentinel-5P Methane dataset and filter by valid date range
2 var methane = ee.ImageCollection("COPERNICUS/SP/OFPL/L3_CH4")
3   .filterDate('2019-01-01', '2019-12-31') // Use a valid date range
4   .select('CH4_column_volume_mixing_ratio_dry_air');
5
6 // Load the uploaded dataset as a FeatureCollection
7 var locations = ee.FeatureCollection('projects/ee-hussnain/assets/df_locations');
8
9 // Define a region of interest by creating a bounding box around the points
10 var region = locations.geometry().bounds(); // Get the bounding geometry of all points
11
12 // Check if the filtered dataset has any images
13 if (methane.size().getInfo() > 0) {
14   // Calculate the mean CH4 concentration over the time range
15   var ch4Layer = methane.mean();
16
17   // Sample the CH4 data at the given locations
18   var sampledPoints = ch4Layer.sampleRegions({
19     collection: locations, // Points or polygons to sample
20     scale: 7000, // Match Sentinel-5P resolution (7x7 km)
21     geometries: true // Include geometry in the output
22   });
23
24   // Print sampled CH4 values to the Console
25   print('Sampled CH4 Data:', sampledPoints);
26
27   // Optional: Visualize the CH4 concentration and locations on the map
28   Map.centerObject(locations, 6); // Center map on the region
29   Map.addLayer(ch4Layer, {min: 1750, max: 1950, palette: ['blue', 'green', 'yellow', 'red']}, 'CH4 Concentration');
30   Map.addLayer(locations, {color: 'black'}, 'Locations');
31
32   // Export the sampled data as a CSV file to Google Drive
33   Export.table.toDrive({
34     collection: sampledPoints,
35     description: 'CH4_Values',
36     fileFormat: 'CSV'
37   });
38 } else {
39   print('No methane data available for the specified region and time range.');
```

The right sidebar shows the 'Inspector' and 'Console' tabs. The 'Console' tab displays a welcome message: 'Welcome to Earth Engine! Please use the help menu above (?) to learn more about how to use Earth Engine, or visit our help page for support.'

CODE:

```
// Load Sentinel-5P Methane dataset and filter by valid date range
var methane = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_CH4')
    .filterDate('2019-01-01', '2019-12-31') // Use a valid date range
    .select('CH4_column_volume_mixing_ratio_dry_air');

// Load the uploaded dataset as a FeatureCollection
var locations = ee.FeatureCollection('projects/ee-hussnain/assets/df_locations');

// Define a region of interest by creating a bounding box around the points
var region = locations.geometry().bounds(); // Get the bounding geometry of all
points

// Check if the filtered dataset has any images
if (methane.size().getInfo() > 0) {
    // Calculate the mean CH4 concentration over the time range
    var ch4Layer = methane.mean();

    // Sample the CH4 data at the given locations
    var sampledPoints = ch4Layer.sampleRegions({
        collection: locations, // Points or polygons to sample
        scale: 7000, // Match Sentinel-5P resolution (7x7 km)
        geometries: true // Include geometry in the output
    });

    // Print sampled CH4 values to the Console
    print('Sampled CH4 Data:', sampledPoints);

    // Optional: Visualize the CH4 concentration and locations on the map
    Map.centerObject(locations, 6); // Center map on the region
    Map.addLayer(ch4Layer, {min: 1750, max: 1950, palette: ['blue', 'green',
'yellow', 'red']}, 'CH4 Concentration');
    Map.addLayer(locations, {color: 'black'}, 'Locations');

    // Export the sampled data as a CSV file to Google Drive
    Export.table.toDrive({
        collection: sampledPoints,
        description: 'CH4_Values',
        fileFormat: 'CSV'
    });
} else {
    print('No methane data available for the specified region and time range.');
```