Assimila Technical Report

Technical Report for Suriname's Counterpart, Based on In-person Training and Virtual Learning Sessions

Gerardo Lopez Saldana & Alex Cornelius

June 2024

Global Partnership for Sustainable Development Data

Assimila

Acronyms and Abbreviations

Abbreviation	Description
ASGM	Artisanal Small-Scale Gold Mining
AOI	Area of Interest
EO	Earth Observation
GEE	Google Earth Engine
NDVI	Normalised Difference Vegetation Index
VM	Virtual Machine

Table of Contents

List of Figures

Introduction

Artisanal small-scale gold mining (ASGM) is an extremely harmful practice, often resulting in pollution – particularly, mercury contamination – and deforestation of ancient rainforest. This practice is prevalent in Suriname and is difficult to monitor due to the remoteness of the affected regions. Remote sensing, which is the acquiring of information from a distance via sensors on satellites and aircraft, can excel at detecting events and practices in hard-to-reach areas. This is due to large-scale data collection practices, done in a timely and repeatable manner, meaning vast amounts of salient data can be collected over large remote regions. Remote sensing data collected from orbiting satellites can be used to characterize the land surface at scale, leading to invaluable insights about current landcover trends and changes. Currently, efforts are underway to leverage earth observation (EO) data to derive robust methodologies to model the landcover of Suriname – including in areas where ASGM is practiced.

Members of the scientific community within the country are working on EO static datasets provided by the environmental consulting company Assimila to derive landcover maps to help identify areas of changing ASGM prevalence. The detection of exposed bright soil is often an indicator of mining activity. Assimila also consulted with members of the scientific community in Suriname on best practice techniques and has provided preset workflows for participants to follow, which removes some of the technical overhead. Assimilla also continues to monitor the public code repository that users have access to, where the codebase can be downloaded and monitored, as well as raise issues.

This report details the technical work done by the Assimila team to further the project's goals. Firstly, it presents the data used in the workflow and its download. Then it discusses the methodology used to create a "clean" dataset, which involves removing pixels with cloud contamination and generating spectral index data and subsequent multitemporal composites to act as features in the machine learning model used. The third section outlines the machine learning workflow used to train and implement the model over a wide spatial scale. The final section describes the Jupyter Hub, which participants can access to carry out this work.

1: Model data

To detect ASGM, we focused on using freely available data from Copernicus Sentinel-2 Mission, which is a medium-resolution, multispectral imaging mission supported by two near-polar orbiting satellites. This mission has been operational since March 2017, and is used by the Remote Sensing community for a variety of conservation and environmental monitoring purposes. This data benefits from a spatial resolution of 10 meters, which importantly is sufficiently high to detect the ASGM landcover. ASGM landcover often covers small areas, so other related EO platforms are not capable of detecting the signal of ASGM as spectral data will be contaminated with surrounding landcover types, i.e. rainforest. Even above the often clouded Suriname landscape, Sentinel-2 detects some cloud-free data due to its low global revisit time. With both Sentinel 2A and Sentinel 2B (the two satellites of the mission) in their near polar orbiting arrangement, it is possible to get new data every ~six days, meaning that participants in this project have a significant data repository from which to derive their landcover maps.

Sentinel-2 also has a wide variety of spectral bands able to detect features across a broad range of the electromagnetic spectrum. The majority of the bands are found in the visible and near-infrared regions of the electromagnetic spectrum, which is advantageous given that the majority of landcover changes due to ASGM are evident from vegetation cover, which is particularly sensitive to this type of energy.

Assimila has significant experience with downloading, preprocessing, and handling geospatial data, with a particular focus on Sentinel-2 data. Therefore, we adapted downloaders from existing Assimila code to create bespoke Jupyter Notebooks capable of retrieving this data from the Google Earth Engine (GEE). These notebooks are available at the publicly accessible Github at the following address: https://github.com/AlexCornelius/EO4ASGM/blob/main/sentinel_2_downloader.ipynb.

These downloaders were built to be easily accessible to people with little experience in handling geospatial data, and therefore were designed to use as few external libraries and tools as possible. The only input the notebook requires is a geographically explicit JavaScript Object Notation file outlining a bounding box of the area of interest (AOI) and a year variable (for example, 2019), so as to minimize the amount of data gathered from the GEE at one time. The script establishes an Application Programming Interface

connection to the GEE and requests all Sentinel-2 bands available for the AOI, then sends a request to transfer the data to the Google Bucket, or Google Drive, of choice. Processing data this way reduces the storage overhead to acquire data and leaves the majority of the processing on the GEE. The data is then transferred from the Google Bucket to a local machine, where it is available to participants.

The Assimila team ran this notebook on behalf of the participants so that they could focus their efforts on the scientific content of the project, rather than the data acquisition itself. Data was downloaded for the years 2019 and 2022, which were selected for the project because both these years have a complete catalog of data, and the gap between them would highlight areas of significant landcover change – for example, from forest cover to the deforestation associated with ASGM.

The AOI was delineated collaboratively with participants. The one that was identified had a variety of landcovers so that participants would have to establish a comprehensive modeling framework, contained clear signs of ASGM, and included significant rainforest as well as the northern tip of Lake Brocopondo, so that water landcover was included in the modeling. This area encompasses 674 square kilometers, which required 13 gigabytes of data to cover the two study years.

Figure 1 Location of the area of interest (green dotted line)

2: Creating a "clean" Sentinel-2 dataset, spectral indexes, and temporally aggregated data

Due to shorter wavelengths of Sentinel-2 band data, when a pixel is covered by clouds the signal of the surface is obscured by the clouds' spectral signature. This means that to obtain an accurate understanding of the landcover with Sentinel-2 data, the user needs to be able to automatically eliminate clouded pixels from the analysis. Many methodologies can accomplish this task, including the multitemporal "difference to reference" methodology presented at the original workshop. This is a classical technique in Remote Sensing in which clouds are identified by finding significant deviations from the "reference" reflectance. The "reference" is found by using the full timeseries of blue, green and red reflectance and finding the mean of the lowest 10 percent of values, which represents the normal cloud-free behavior of the pixel.

But due to the large spatial scale of the AOI, implementing this methodology over a wide spatial scale would be extremely computationally intensive. As such, the clouds are eliminated with a secondary dataset. This is the "s2cloudless" layer, which is the output of a machine learning regressor trained to estimate the probability that a pixel is clouded (https://github.com/sentinel-hub/sentinel2-cloud-detector). This product is available through the GEE and has been generated for all Sentinel-2 data layers, so whenever users download the raw spectral data, they can also download the s2cloudless layer, which will have the same dimensions as the Sentinel-2 data. This is useful for participants, as whenever they open any Sentinel-2 data, they can also open the s2cloudless layer, create a cloud mask, and set the clouded values in the Sentinel-2 data to "not-a-number". This means clouded values are excluded from any processing and will not affect the results of the analysis.

This method also allows users to experiment with customizing their cloud-masking criteria, where the s2cloudless values range from 0 percent to 100 percent (100 percent being the condition where the pixel is entirely clouded), so participants must set a threshold for what probability of cloud they are willing to include in their dataset. Too high a threshold, and some areas will be entirely masked out; too low and there will be

a significant influence of clouds on the pixels' temporal behavior. This threshold should stimulate scientific discourse among the participants as to which threshold is the most appropriate, where the default is set to 20 percent in the example notebook.

An example of this model output is shown in Figure 2, where the left panel shows the Sentinel-2 Band 2 (blue band) reflectance and the right shows the s2cloudless output for the same area. It is clear that bright clouded regions in the south of the image correspond with regions of high cloud probability.

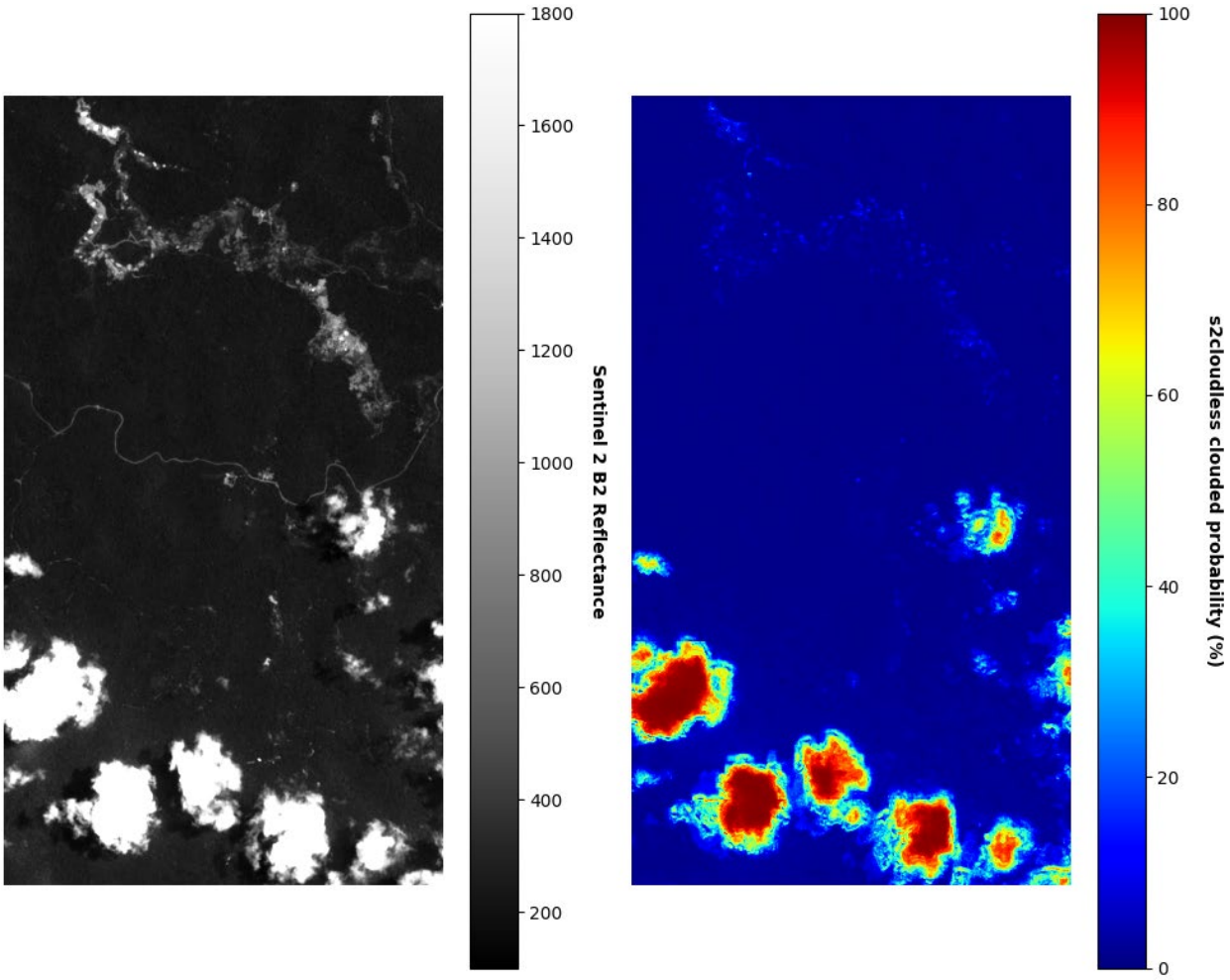

Figure 2. Left: Example reflectance data for the AOI, which includes partial cloud cover. Right: s2cloudless probability corresponding layer.

Once the data has been cloud masked, the next step is to create spectral indexes. The reflectance data itself is a powerful discriminator between different landcover types, but often further information is obtained by comparing the reflectances between spectral bands. For example, a unique feature of the spectral response of vegetation is the high reflectance of near-infrared data compared to the visible light. This can be exploited in the Normalized Difference Vegetation Index (NDVI), which is a spectral index that compares the reflectance of the red band of EO data to the near-infrared data. NDVI will return high values for pixels that conform to the known reflectance response of vegetation.

Therefore, a Python function was provided to participants so that they could easily generate their own spectral indexes, with a few examples included to precompute common spectral indexes. This function was written to ingest a Python dictionary with the spectral data contained in each directory of the library as a 3D numpy array, and then perform common matrix manipulation on these arrays to generate spectral indexes. The output of this calculation is then added to the input dictionary under a user-specified key. This enables participants to research their own spectral indexes and add whichever ones they think would be valuable to the function and help discriminate between landcover types. For example, indexes based around identifying water, like the Normalized Difference Water Index, would be helpful.

Once the participants generate their spectral indexes, the next task is to aggregate the 3D datasets of spectral indexes and spectral reflectance values into 2D composites in the time dimension of the data. The aggregations summarize the temporal behavior in a variety of ways for each pixel. The most common is the average value throughout the year, where different landcover types will have different average values. For example, the exposed bright soil that is often a feature of ASGM will have a much higher average value in the visible Sentinel-2 bands than water and forest cover. The process of aggregating the data is written into another piece of code for use by the participants, where the input to the function is a 3D array, which could be any of the datasets the participants generate themselves. The 3D array is aggregated over time using any of the possible "nan-friendly" numpy functions along the time axis, where the participants can pick aggregators like np.nanstd, which returns the standard deviation of the pixels' behavior across time. The documentation within the function shows where users can read about different aggregators, and participants are encouraged to experiment.

Often, different landcover types exhibit distinct temporal behavior, which can be summarized by different temporal aggregators. For example, vegetative landcover, e.g. rainforest, will vary its reflectance over the year given the phenology and seasonality of the plant. This means the degree to which reflectance changes can be summarized by the pixels’ standard deviation. This is distinctly different from other landcover types, like roads and exposed concrete, which will generally exhibit very little change over the year and have a low standard deviation in a pixel’s reflectance. Hence, capturing temporal behavior with a variety of different temporal aggregators can be invaluable in landcover classification workflows and increase the types of data available to distinguish between landcover types.

3: Machine learning model building

Once the participants have run the example notebook, they will have successfully generated cloud-free Sentinel-2 datasets, exploited spectral relationships by generating spectral indexes, and temporally aggregated these to create 2D composites for all variables. The next step is to transform these composites into a meaningful landcover classification. To do this, the participants must use the training data supplied by the local partners in Suriname - Stichting Bosbeheer & Bostoezicht (Forest Management & Forest Supervision Foundation, termed SBB). This data is a set of shapefiles containing labeled polygons that delineate the extent of the three main landcover types found in the AOI: water, forest, and ASGM. This is an invaluable dataset that makes the whole workflow feasible, as when using EO data it is often very difficult to find accurate training data.

SBB supplied the shapefiles in a multi-feature shapefile for each individual year. The first job was to split these multi-featured shapefiles into shapefiles containing a single feature each. Doing so reduces the amount of data opened when training the model. Given the size of the AOI, cutting down the memory usage is very important. This task was carried out by members of the Assimila team, and the single-feature shapefiles were added to a common directory that all the participants could access. The next step was creating code in the example notebook to use these shapefiles to open the Sentinel-2 data that all participants could run. The code to do this was a loop that iteratively used each of the features for a single study year to open each of the Sentinel-2 data files. When using the shapefiles to open the data, the code will only open data contained within each polygon. This means that if the shapefile delineates water landcover, the returned Sentinel-2 data only contains data covering water landcover.

A visualization tool has been given to participants to help visualize the “separability” of the spectral features between the different landcover types. The best tools in any machine learning model are features that are distinctly different between landcover types, and this visualization displays a probability density of each of the features for the three landcover types. Features that will be especially effective for the classification model are ones where the histograms from each of the three landcover types take up different regions of the feature space. Figure 3 shows an example of this visualization, where the blue, green, and orange histograms represent the distribution for water, forest,

and ASGM landcovers, respectively. An example of a very effective feature is the NDVI mean feature (bottom left panel), where the three histograms for each landcover are very separated and uniquely distributed.

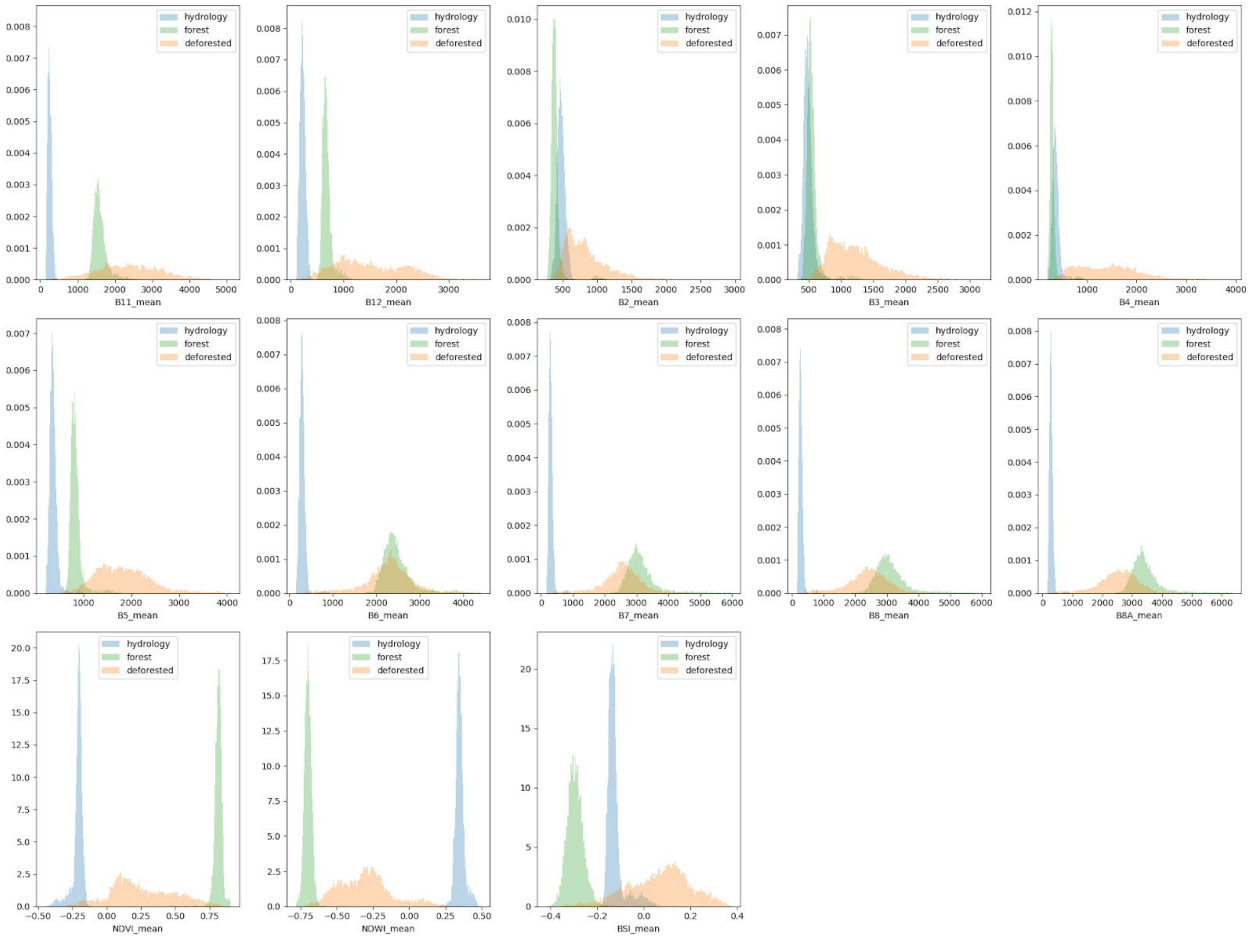

Figure 3. Feature distribution visualization tool available for participants to show the separability of different features available to the machine learning.

Once the data has been opened with these shapefiles, using the code provided, the participants can easily calculate the spectral indexes and aggregate the data for each of the individual features. The next task is to build a training dataset and a validation dataset. The training dataset is the majority proportion of the compiled data, normally 70 percent, that a machine learning model will use to build the statistical relationships between the input variables and the target variable, i.e., the landcover class. The validation dataset is what the trained machine learning model will be tested against to check its validity and accuracy, and is completely unseen data to the model so that the

results are a representation of its accuracy when applied to new data. The training and validation datasets are derived by randomly sampling the complete dataset, where the code to do this is given in the example notebook.

That last stage of the model building is to train a machine learning model on the training dataset. The implementation of two models is given in the example notebook, where the contrast between the two models will yield interestingly differing results. The first example given is a single decision tree provided with the Sklearn library. This is the most basic of supervised classifiers and will iteratively split the feature space to increase the accuracy of the classification. The second model is a gradient-boosting ensemble of decision trees provided in the library XGBoost. This creates a multitude of decision trees that are trained on a random subset of training data and features to make the final classifier more robust in unseen scenarios. When establishing these models in the example notebook, the parameters are supplied for participants with the intention that they experiment to improve their results and make their models more robust. In addition, code is supplied to test the accuracy of the validation dataset. This measurement is supplied as an accuracy score, which summarizes the average accuracy of the classification for all the classes.

The final step in the entire workflow is to implement the model. A function is supplied to the participants that enables them to implement their model over the entire AOI in a memory-efficient way, so that multiple users can carry out classifications at the same time. To save on memory, the classification is applied onto 500x500-meter chunks iteratively across the AOI, which opens data in piecemeal spatial subsections. The outputs of each chunk classification are stitched together and saved to disk as geotiffs. Each classification represents the annual landcover classification for the study year selected at the beginning of the notebook, i.e., 2019 or 2022. The difference between the classifications represents significant change in spectral signatures indicating that the landcover has changed.

4: Jupyter Hub

It is important for collaborative research that participants have a common environment in which to work, where there is access to data, computation resources available for large processing jobs, and significant disk space for input data as well as participants' outputs. With all of this in mind, members of Assimila set up a Jupyter Hub running on a Google cloud virtual machine (VM). This VM was set up with plenty of storage space and processing capabilities, where a Jupyter Hub is constantly available for participants to login via their unique credentials and run Jupyter Notebooks. This enables participants to experiment by themselves and understand the codebase in their own time. Results written by participants will be saved to their home directories and subsequently exported to their own working environment.

Figure 4 displays a screenshot of the Jupyter Hub, which shows a helpful file explorer on the left-hand side and the main Jupyter Notebook in the center of the screen. This figure shows an example classification for the AOI generated by members of the Assimila team using the example notebook provided.

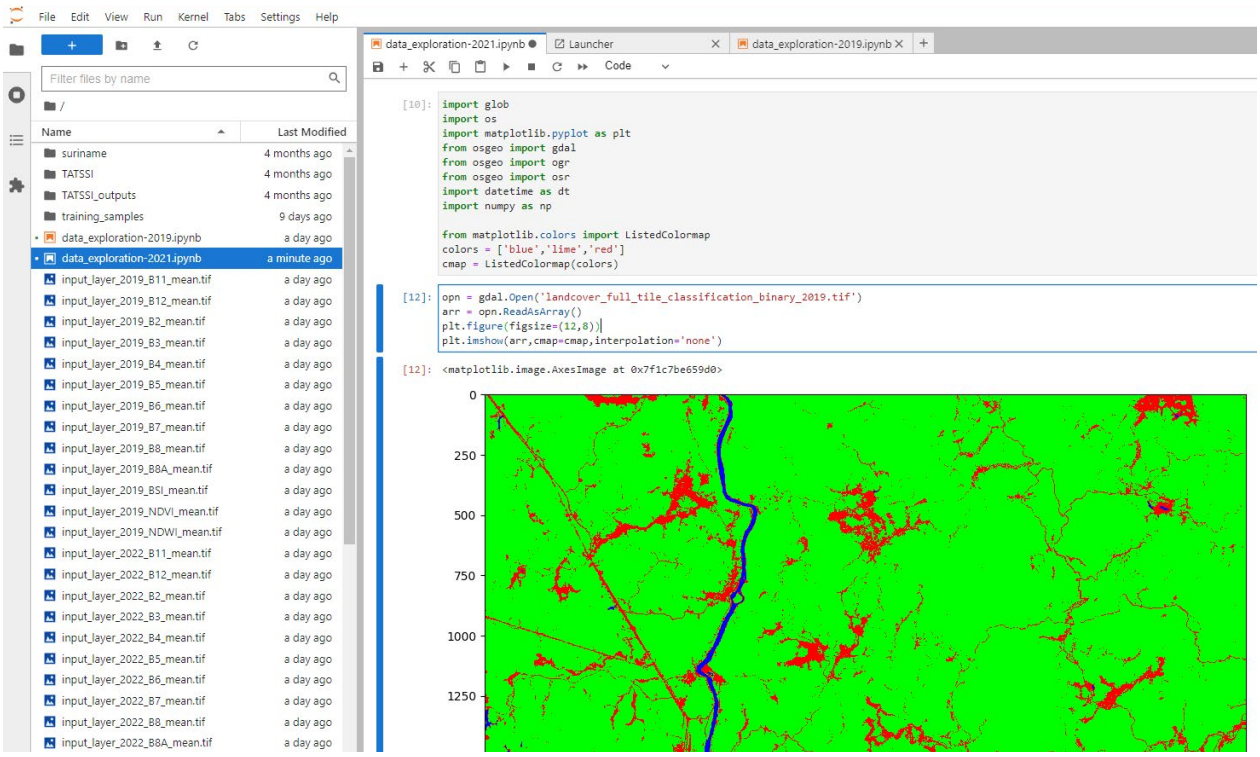

Figure 4 Example screenshot of the Jupyter Hub available to participants.

5: Conclusions

EO and machine learning are invaluable tools when monitoring the spatial and temporal distribution of ASGM. Within this project, freely available Sentinel-2 data was downloaded and made available to local partners in Suriname, alongside helpful code and test scripts, so that landcover classifications could be generated for an AOI in inner Suriname. Participants were able to generate landcover classifications for 2019 and 2022, so that difference analysis could be performed between the classifications. By using these datasets, Assimila and other participants found that there was a loss of ~24km^2 of primary rainforest to ASGM landcover. In addition it was found that the area covered by ASGM increased by 47 percent in the AOI.

Given these concerning statistics, it reinforces the need for continued research to make this sort of analysis consistent and systematic, so that an accurate understanding of the changing landscape can be made. ASGM is a significant and increasing threat to primary rainforest in Suriname, where repeat surveys of the change in landcover should be carried out. The methodologies and techniques used in this analysis are now publicly available and have been written in such a way that makes the workflow scalable. This means the spatial area of the analysis can easily be increased by simply increasing the spatial area of the initial AOI. With this in mind, this work represents a promising new tool in monitoring ASGM by increasing the technical capacity of local researchers in the field of EO data handling, EO manipulation and machine learning.

This document is a product of the work led by the Global Partnership for Sustainable Development Data. With funding from the Islamic Development Bank and the Inter-American Development Bank.

For more information:

data4sdgs.org

info@data4sdgs.org

Global Partnership for Sustainable Development Data

Assimila