Asset: Automated pipeline for real time map generation

Date: April 2023

1. Content

This asset is provided by the following items:

- Modules: Folder that contains a python script with the function needed for map generation.
- Plots: Folder that contains some plots from La Rioja, Spain.
- Static: Folder that contains the file index.htlm to display the frontend of the website.
- Main.py: Python file, the main script that must be run to display the website.
- Plot.geojson and raster.plk are auxiliar files overwritten when the user interacts with the website.

2. Automated pipeline. An overview of the files.

2.1. Considerations

These files have been run inside of a server where satellite pictures are uploaded. The user will need to adjust the path for their local environment and download the Sentinel-2 images (in the code loaded from SAFE folder).

2.2. NVDI

Normalized Difference Vegetation Index (NDVI) uses the NIR (Band 8) and red (Band 4) for calculations

$$NVDI = \frac{NIR - Red}{NIR + Red}$$

NDVI always ranges from -1 to +1. Healthy vegetation (chlorophyll) reflects more near-infrared (NIR) and green light compared to other wavelengths. But it absorbs more red and blue light. Negative values correspond to water or buildings, values close to +1, means dense green leaves and close to zero, there aren't green leaves, and it could even be an urbanized area.

The process followed to calculate NDVI for every plot is:

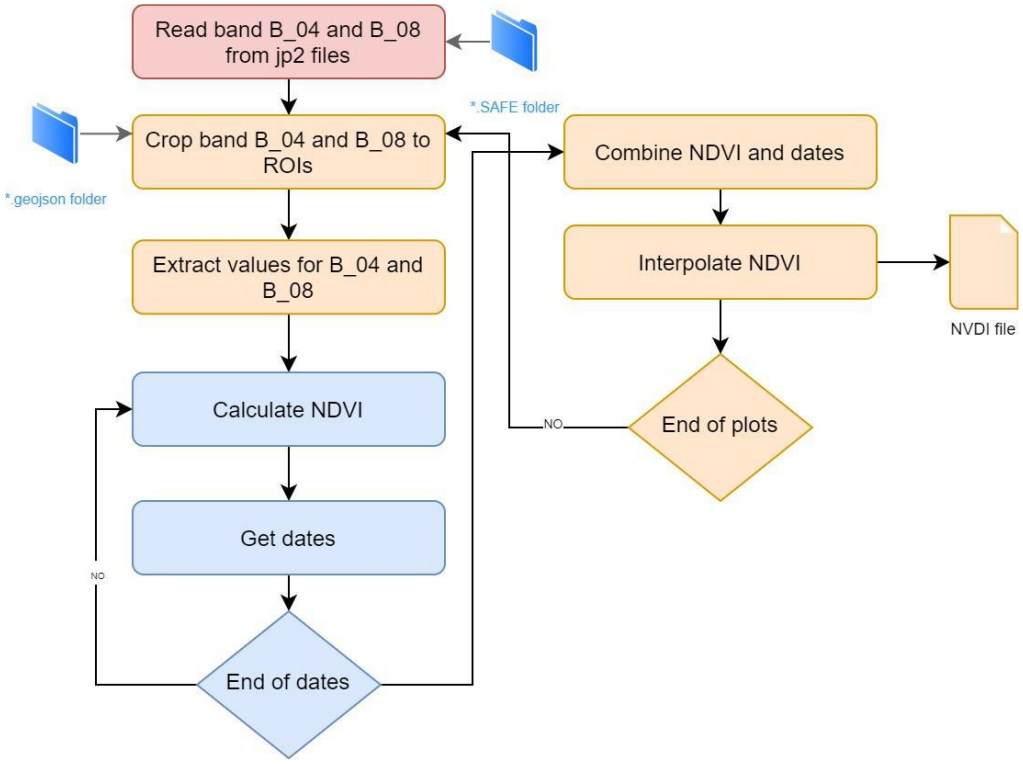

Figure 1. Flowchart for NVDI Calculation

An interpolation stage is needed to complete all dates as merging processes need all dates with a numerical value. The dates conversion was made using following fixed dates:

- Day 2018-04-19 is *day_index = 0*
- Day 2022-04-18 is day_index = *daysbetween(2022/04/18 - 2018/04/19) = 1460*

Interpolation is made using linear interpolation *interp* from *numpy* library.

2.3. Plots situation for imaginary selection

Plots included in the asset are selected from vineyard with two grapes variety:

- Tempranillo: 7 plots (P101, P102, P106, P111, P117, P119, P93), total area: 43.576 m2
- Garnacha: 4 plots (P110, P109, P223, P81), total area: 84.413 m2

The situation of these plots is:

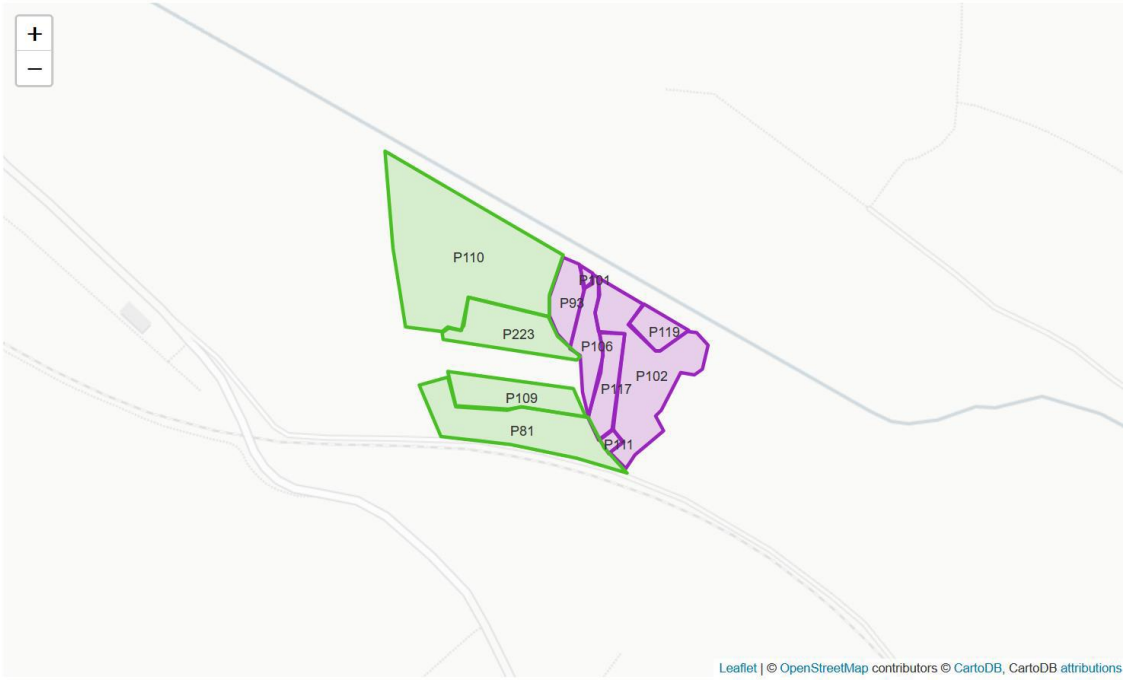

Figure 2. Map of vineyards plots using geojson

All coordinates of the *region of interest* (ROIs) are located in tile 30TWM, as shown in the figure:

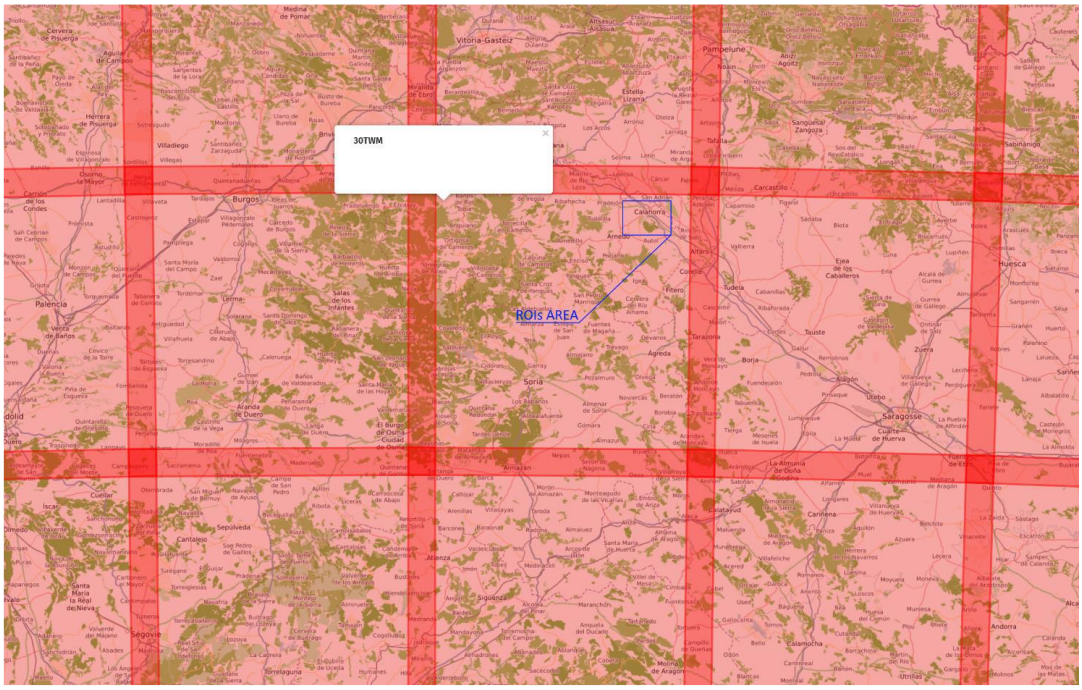

Figure 3. Tiles for north of Spain and tile 30 TWM

The script has been generated with the selection of tiles in SAFE format for 4 years (maximum available) from 2018-04-19 to 2022-04-18, and a cloud percentage limited to 15% and manually checked and chosen.

2.4. Website appearance

Once the browser is loaded (after running *main.py*), the website will look like this:

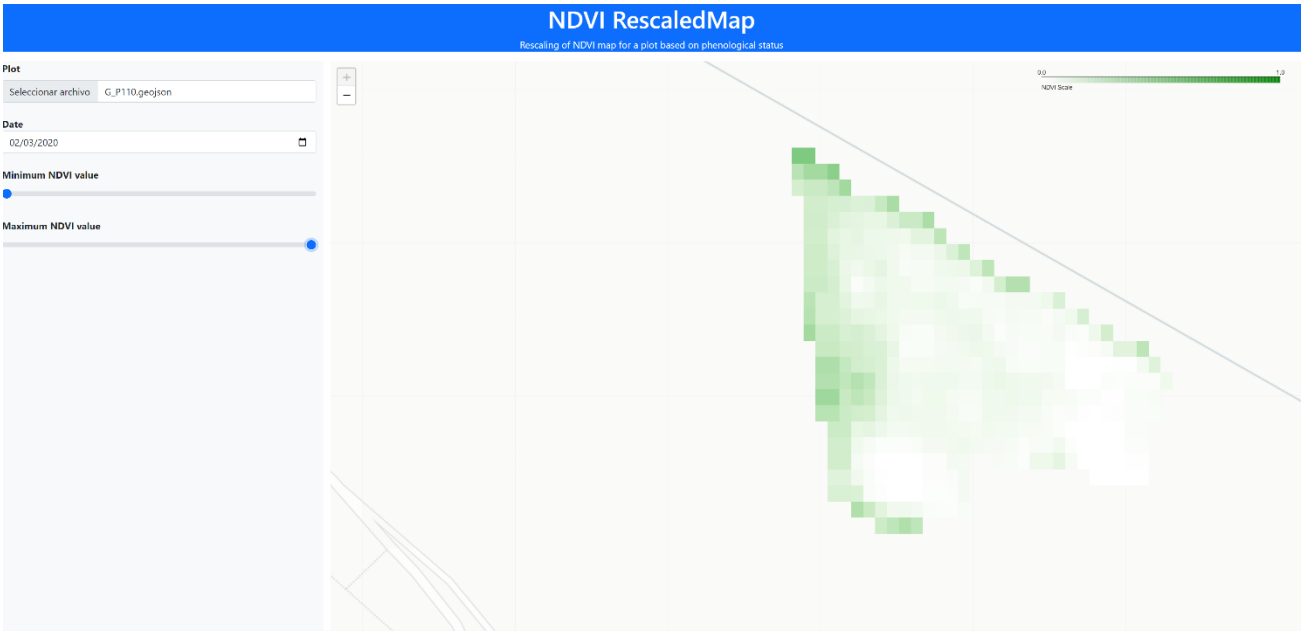

Figure 4. Website

On the left side, there are the following items to interact:

- Plot section: it allows to load a geojson file, available in /plots/geojson/
- Date: a selector to choose the date for the NVDI calculation.
- NVDI scroll bars: these scroll bars allows to manually adjust the minimum and maximum NVDI value from 0 to 1.

On the right side, the user will visualize the map for the crop defined on the geojson filled by the NVID scale.

2.5. Website interaction. An example

For this example, the plot G_P110.gejson and date 2019-12-02 are selected; minimum and maximum NVDI values remain as default, from 0 to 1:

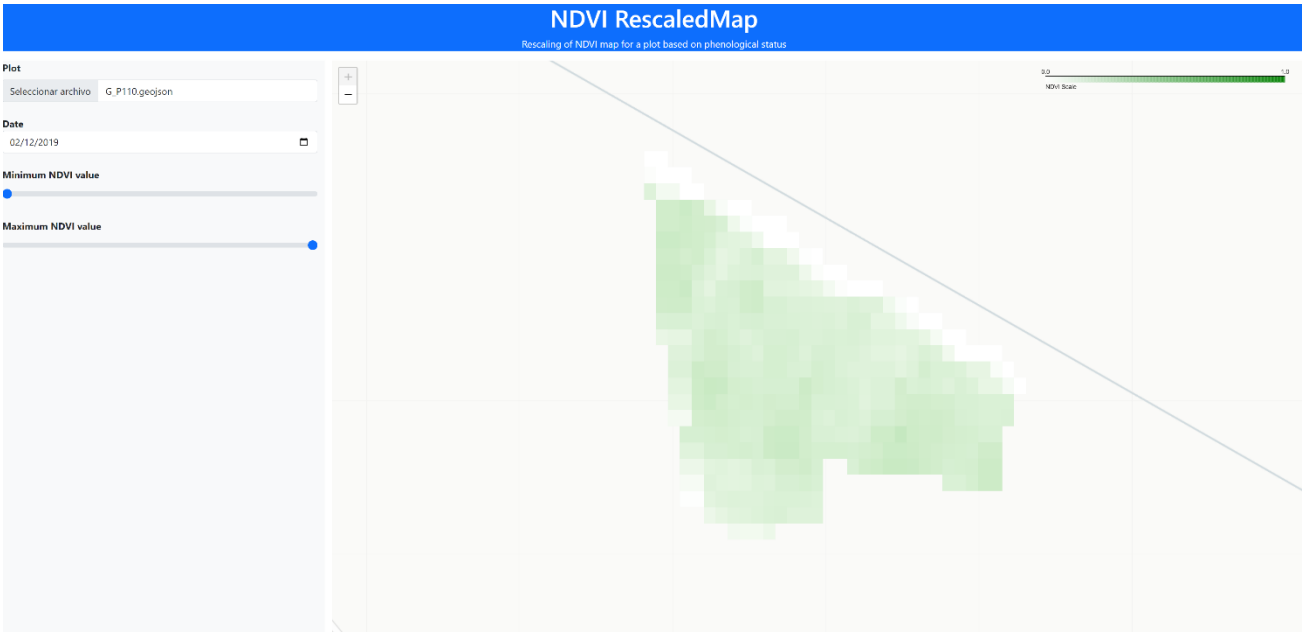

Figure 5. Selection of T_P106.geojson plot

Then, the minimum NVDI is modified to 0.05 and maximum NVDI to 0.35. As the it is shown in Figure 6, the colours on the plot have changed to adequate to the manual NVDI scale selected.

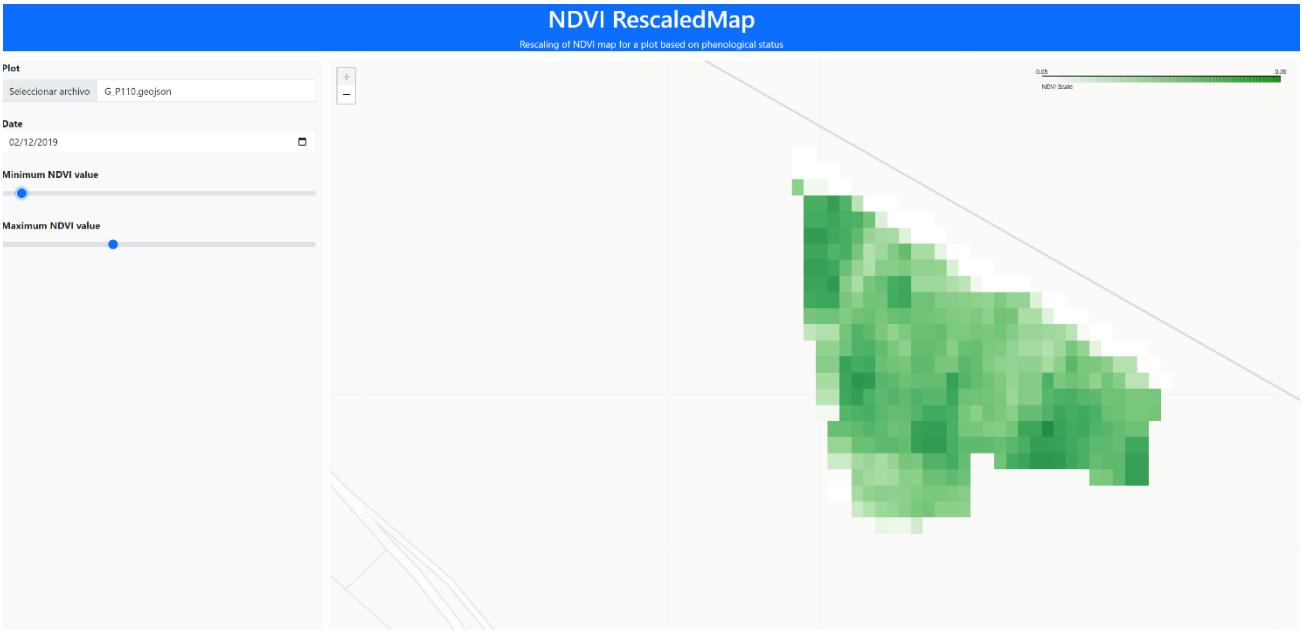

Figure 6. Rescaling NVDI values

Finally, keeping the selected scale, another date (2020-03-02) is chosen to analyse the crop evolution. In this picture. In this season, the crop has less vegetative activity what entails a lower NVDI value (coloured whiter).

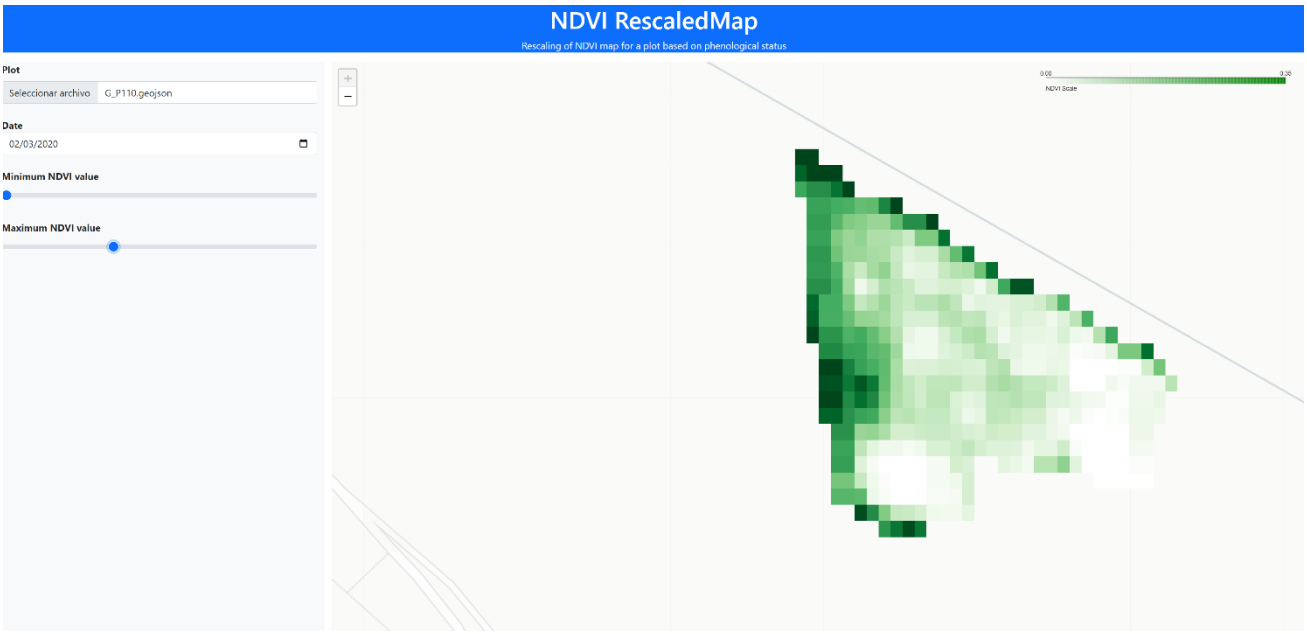

Figure 7. Selection of different date

2.6. Backend

This website has been programming through a python library called *Flask*. This library allows website developing through python code, which will help to include the solutions achieved in the previous deliverables.

This library is used in the file *main.py* that has been run in the beginning. When the file is launched, it configures in the port *5000* according to the file *index.htlm* to display the items on the screen.

Whenever there is a change on any of the items of the website, the raster of the plot is generated and the NVDI modelling is recalculated to be displayed on the map.

2.7. Frontend

The website interface is defined by the file *index.htlm* that has been programming through htlm. Besides the visualization configuration is programmed by htlm, some python functions are also needed and implemented using the functionality of *PyScript*.

For the configuration and design of the web, the CSS.Boostrap is used (https://getbootstrap.com/docs/3.4/css/). Moreover, the iterative elements for the selection of plot, dates and NVDI range are defined and located. This file is continually running, whenever there is a change, defined as an iteration with any of the selectors or slider items to set a date, NVDI, or plot values; there is an update in the visualization. Updates includes the following tasks:

- Upload a file: the map is loaded according to the geojson readed.
- Date change: the satellite picture for the nearest date available is selected. This picture is cropped and modelling with the NVDI to reload the map.
- Slider change: the map coloured and NVDI is recalculated and reloaded on the map according to the slider bar values. This scrolls bar can be settled with values from 0 to 1 by a 0.05 step. If the minimum value is scrolled is bigger than the maximum, this second one is automatically set at a value 0.1 times higher than the selected minimum.

As a result, once the server is called, it will consider the current information given by the interactive items in the web, to calculate NVDI model saved on file "raster.pkl", a pickle format with the information needed for updating the map.