The python. ipynb provided are organized as follows:

1 – Image_Processing.ipynb

In this step the plants under consideration are isolated using two different methods: Automated Editing and Semi-Automated Editing. Using multispectral images, the NDVI transform is applied, and the vegetation is highlighted. The user can either apply Automated Editing and Semi-Automated Editing methods, both of which isolate the plant under consideration using two different approaches. Finally, for each tree the pixel coordinates are saved that belong to that tree.

2 – Divide_Image.ipynb

Following the first step, the tree data points are loaded into the code where two different algorithms will be applied to divide trees into subsections. The first approach will be the grid subdivision, a method that divides the tree into a 3x3, 2x2 or 1x1 grid depending on the number of points that belong to the tree. In the second approach k-means clustering is applied dividing the tree into Vornoid cells. For both approaches the subdivided regions are numbered and the image of the subdivided plant is saved along with the pixels datapoints that belong to the tree, assigning to each pixel its respective region number.

3 – Check_Divided_Image.ipynb

Verifying the data produced and saved in the previous step to check if there are any errors or discrepancies in the data generated.

4 – ComputeVIs.ipynb

Computes and saves vegetative indices of each of the subregions for every plant in the dataset. Saves the computed data in a *csv* file.

5-plant_data.ipynb

Plots the sick and healthy ratio of the label applied, plots the distribution of each vegetative index relative to the label applied, plots the correlation matrix between the different vegetative indices computed.

6 – Classification_models1.ipynb

Applies Random Forest, Logistic Regression and K-nearest Neighbors on the dataset and shows the results.

7 – NN_Divide_Images.ipynb

Divides images into the pre-defined regions of step 2, the saved images will serve as the input of the CNNs applied.

8 – Save_NN_rand.ipynb

Randomly shuffles the images computed in the previous step and save them into separate folders according to their applied labels. The saved images will be divided between a training set, a validation set and a test set.

9 – CNN.ipynb

Applies CNNs to the saved images from the previous step. ResNet50 and DenseNet121 are applied.

10 – Results_Histogram.ipynb

Shows the results of the machine learning algorithms applied in a histogram.

11 – Geojson.ipynb

Organizes the data of each shooting day into a *geojson* file. The health status and geographical coordinates are added for each tree.

NB: This code was executed on **Google Colab**. The directories used are linked to a specific google drive account. Consider changing the directories before running the code.