OBJECTIVE: prediction of potential anomalies in crops

Procedure:

1. **Training**:
    a. Extraction of Vegetation Indices on fields with a validated anomaly
    b. Extraction of Vegetation Indices on fields with not-detected anomalies
    c. Ingestion of datasets for training
    d. Training and selection of model
2. **Prediction**:
    a. Extraction of Vegetation Indices on fields to detect status
    b. Prediction of status based on trained model

Datasource: Copernicus Sentinel-2 satellite images
Vegetation Indices: SR, NDVI, GNDVI, NDRE, modNDRE, EVI, EVI2, PVR, GCI, RECI, TVI, MTCI, LCI, TCVI, GARVI, SIPI1, SIPI2, MCARI, ARVI,OSAVI.

Required applications:

- **Anaconda**: Distribution of Python and R programming languages for scientific computing, suitable for Windows, Linux, and macOS, that aims to simplify package management and deployment. (www.anaconda.com)
- **Python Notebook**
- **Orange Data Mining**: Open source machine learning and data visualization application. License GPLv3. (www.orangedatamining.com)
- **Google Earth Engine** account

Use cases:
Case 1: Analysis of fields with potential PSA occurences (based on trained datasets)
Procedure:
a. Extract field information (video)
b. Launch Prediction Workflow (video)

Case 2: Analysis of other potential anomalies. Required new trained datasets
Procedure:
a. Extract historical field information related to the crop anomaly and healthy fields (video)
b. Launch Training Workflow (video)
c. Extract Field information for new prediction (video)
d. Launch Prediction Workflow (video)

Python Notebook required modules: geemap, ee, numpy, eemont, csv, os, io, requests, pandas, osgeo

- **geemap**: is a Python package for interactive mapping with Google Earth Engine (www.geemap.org). To use geemap, is required a Google Earth Engine account (https://earthengine.google.com/)
- **ee**: Google Earth Engine Python API package
- **eemont**: This package extends Google Earth Engine with pre-processing and processing tools for the most used satellite platforms. (https://eemont.readthedocs.io/en/0.1.7/)
- **pandas**: open source Python package most widely used for data science/data analysis and machine learning tasks (https://pandas.pydata.org/)

Anaconda Environment:

1) Install Anaconda:

Install anaconda or miniconda: The **geemap** package has some optional dependencies, such as GeoPandas and localtileserver.

2) Create a new Environment

It is highly recommended to create a new conda environment to install geemap. Follow the commands below to set up a conda environment and install geemap and, which includes all the optional dependencies of geemap:

conda create -n [environment name]

To **create an environment** in which you can work and install the following packages you can use the file **.condarc** , and putting it into the user folder (naming it ".condarc"), and in the Anaconda prompt the following line "*conda config*".

example of .condarc file:

```
channels:
  - conda-forge
  - defaults
create_default_packages:
  - python
  - ee_extra
  - eemont
  - geemap
  - google-cloud-sdk
  - pandas
  - geopandas
  - numpy
  - spyder
  - spyder-kernels
  - jupyter
ssl_verify: true
```

Note: On windows eliminate the "- google-cloud-sdk" package from the .condarc because is not provided on this channel of anaconda.

Earth Engine Authentication:

After the *ee_extra* and *geemap* installation, you can authenticate on GEE from command line in your environment anaconda prompt, and follow the instructions and enter with your google credentials:

conda activate [environment name]
earthengine authenticate

Google Earth Engine (GEE) packages in python (Anaconda environment)

https://geemap.org/installation/
https://github.com/r-earthengine/ee_extra
https://eemont.readthedocs.io/en/latest/
https://anaconda.org/conda-forge/google-cloud-sdk

Note: In case *ee_extra*, *eemont*, and *geemap* packages have not been installed in the initial Anaconda environment using the .condarc file, they can be installed later from the command line:

Anaconda prompt ->

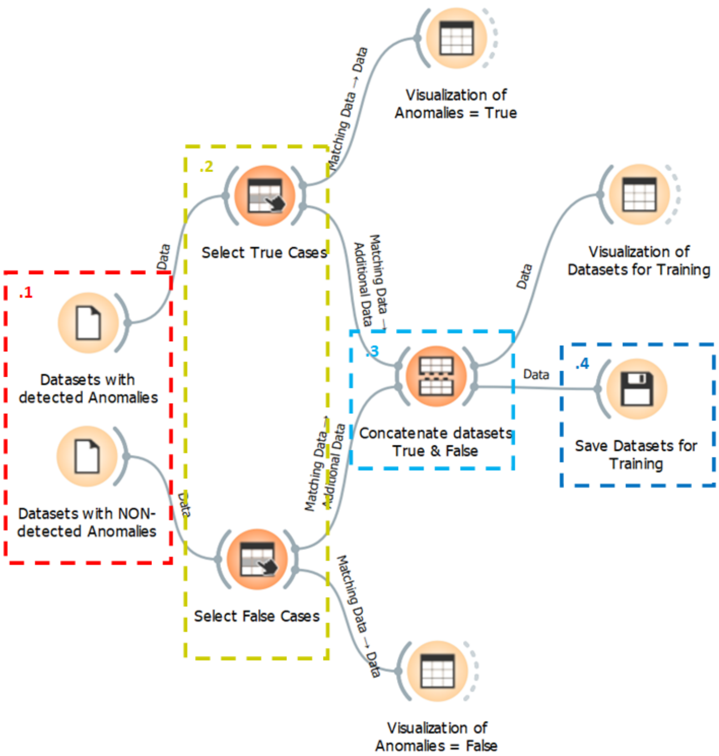
conda activate [environment name]
conda install geemap -c conda-forge
conda install ee_extra -c conda-forge
conda install eemont -c conda-forge
conda install geedim –c conda-forge

Alternatively Install from PyPI: geemap is available on PyPI. To install geemap, run the following command in the terminal: ***pip install geemap, pip install eemont***

Note: In case the command "*earthengine authenticate*" doesn't work because the command 'gcloud' is not recognized, install the google-cloud-sdk package, close all Command Prompts and re-open the environment to proceed with the authentication.

- Linux: *conda install -c conda-forge google-cloud-sdk*
- Windows: follow instructions described on: https://cloud.google.com/sdk/docs/install

Data Preparation Workflow



Procedure:

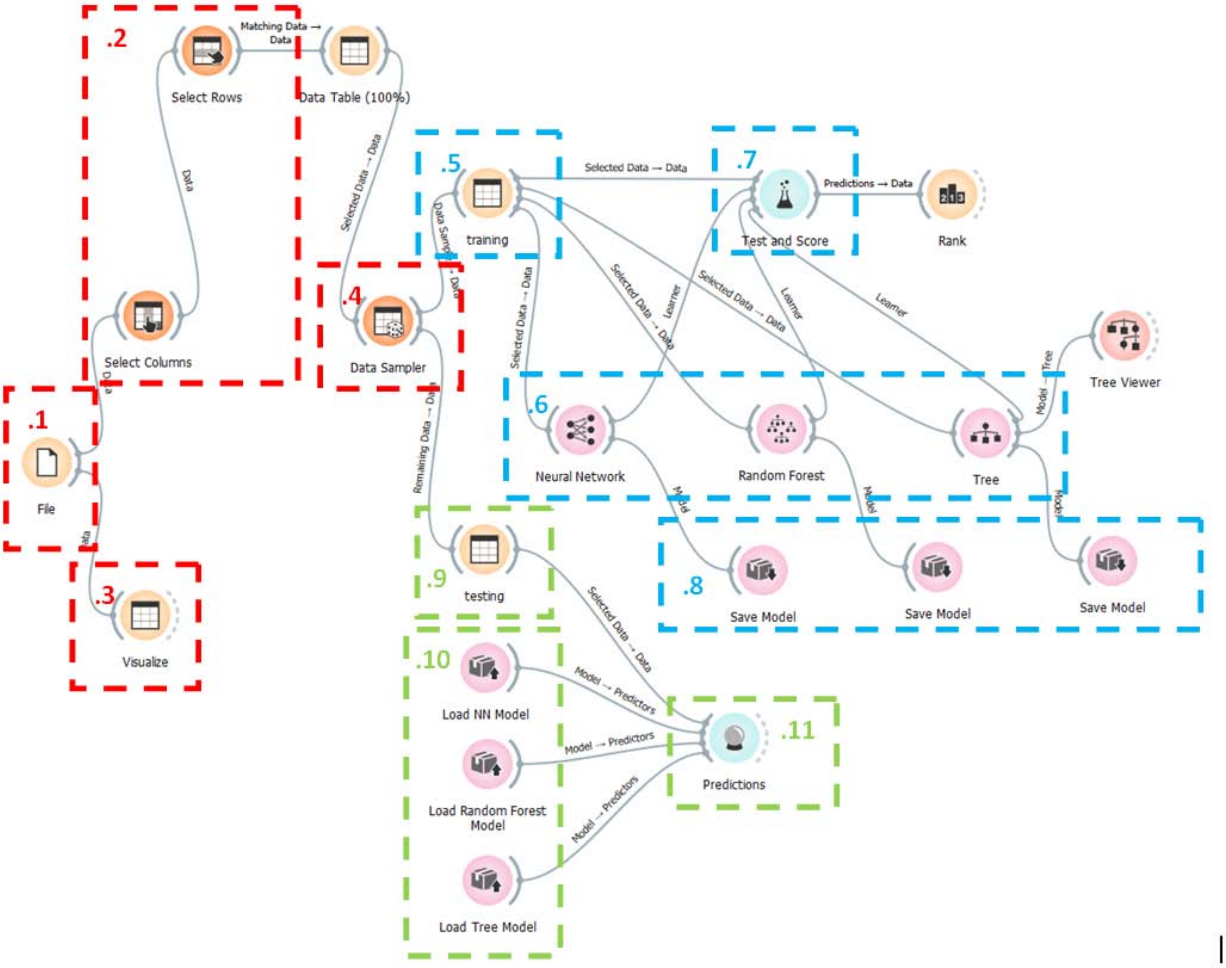
Step 1: Open Orange Data Mining application and load the workflow, or launch it via command line:
*C:\Program Files\Orange3>orange-canvas C:\AnomalyDetection\Orange_Workflows**A_Workflow_dataPreparation.ows***

Step 2: reload the files selected in **1**

Workflow description:

1. Upload of datasets (.csv format) generated with VI Python Notebook (fields_anomaly_true.csv & fields_anomaly_false.csv files)
2. Select particular rows (fields) if necessary
3. Concatenation of True/False datasets into a single File
4. Save final dataset for training “fields_for_training.csv” (required input for the Training Workflow)

Training Workflow



Procedure:

Step 1: Open Orange Data Mining application and load the workflow, or launch it via command line:
*C:\Program Files\Orange3>orange-canvas C:\AnomalyDetection\Orange_Workflows**B_Workflow_training.ows***

Step 2: reload the file selected in **1** (fields_for_training.csv)

Step 3: check the model with the best metrics from **7** (Test and Score)

Step 4: check the model with the best metrics from **11** (Predictions)

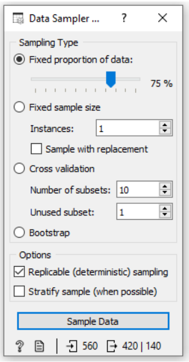
Workflow description:

1. Upload of the file with the information extracted from Sentinel-2 vegetation indices (fields_for_training.csv).

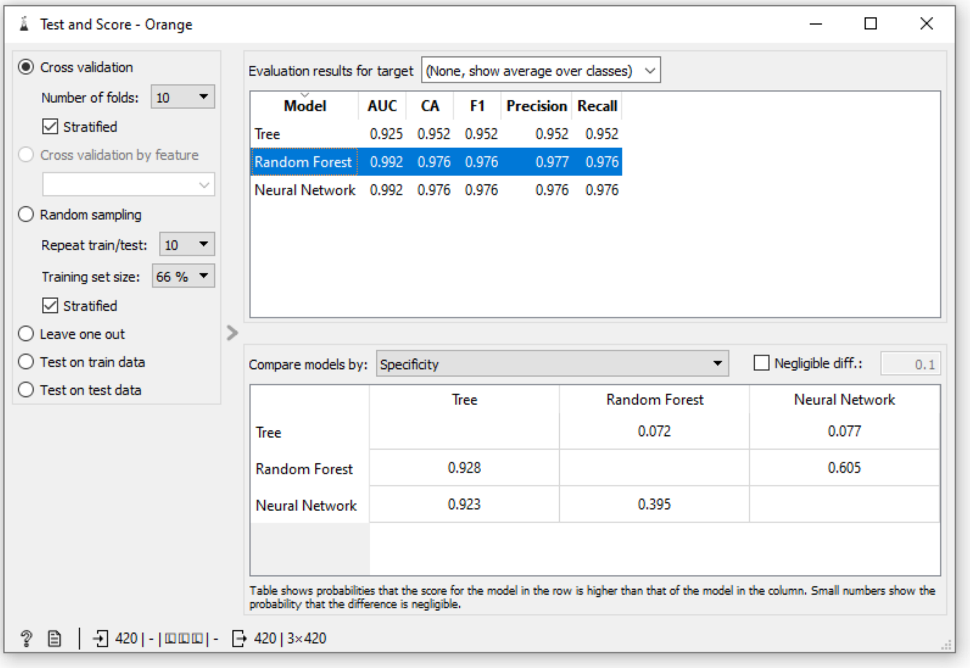
   File format: Comma Separated Values (.csv)

SR_MIN	SR_MAX	SR_MEAN	SR_STD	NDVI_MIN	NDVI_MAX	NDVI_MEAN	NDVI_STD	GNDVI_MIN	GNDVI_MAX	GNDVI_
11,45631027	17,79047585	15,199478	1,488961513	0,839438796	0,893563092	0,875416577	0,01229529	0,722337008	0,765702426	0,7491
6,583038807	17,9093647	13,82984728	2,639996275	0,7362535	0,894232333	0,859906775	0,030305478	0,660797834	0,761727452	0,7268
12,28435993	19,35099411	17,17387962	1,390944948	0,849447012	0,901724696	0,889193821	0,009964646	0,732909918	0,769127488	0,7542
13,47126484	20,06920433	17,93799414	0,971313187	0,861795068	0,905074716	0,894061967	0,006483993	0,745181024	0,77170366	0,7588
13,38043499	19,63036346	18,0174886	1,36346598	0,860922158	0,903055489	0,894187247	0,009053697	0,736554384	0,784152985	0,7561
13,25340557	20,14336967	18,41993021	1,544425258	0,859682679	0,905407667	0,896188999	0,010410047	0,738447785	0,781676412	0,7625
8,544061661	17,16770172	14,28157945	2,508008476	0,790445626	0,889914513	0,864425347	0,028889651	0,703464925	0,766887009	0,7494
9,340995789	18,19682503	15,70631237	2,206005932	0,806595027	0,895816088	0,877528337	0,021110713	0,700245678	0,751011252	0,7358

2. Selection of Columns & Rows of interest
3. Data visualization
4. Data Sampler (for instance a selection of 75% of cases for training and 25% for testing)



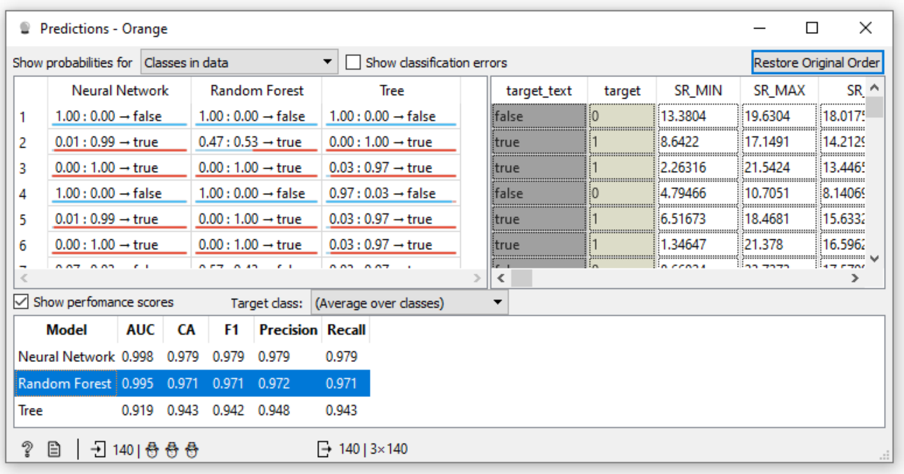
5. Visualization of data for training
6. Definition of model parameters
7. Testing & Scoring evaluation of selected models



- Area under ROC is the area under the receiver-operating curve.
- Classification accuracy is the proportion of correctly classified examples.
- F-1 is a weighted harmonic mean of precision and recall
- Precision is the proportion of true positives among instances classified as positive, e.g. the proportion of Iris virginica correctly identified as Iris virginica.
- Recall is the proportion of true positives among all positive instances in the data, e.g. the number of sick among all diagnosed as sick.
- Specificity is the proportion of true negatives among all negative instances, e.g. the number of non-sick among all diagnosed as non-sick.

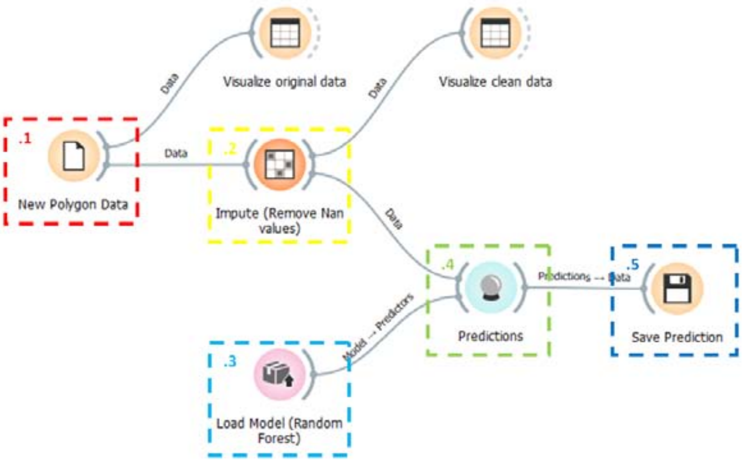
More info regarding the test & score widget at: https://orangedatamining.com/widget-catalog/evaluate/testandscore/

8. Models trained are saved to local folder
9. Visualization of data for testing
10. Load of models saved on step 8
11. Predictions based on saved trained models with a subset selected for testing



12. Selection of the model to be applied for the anomaly detection. In the example for the demo data provided, the selected is Random Forest model.

Prediction Workflow



Procedure:

Step 1: Open Orange Data Mining application and load the workflow, or launch it via command line:
*C:\Program Files\Orange3>orange-canvas C:\AnomalyDetection\Orange_Workflows**C_Workflow_prediction.ows***

Step 2: reload the file selected in **1**

Step 3: check the selected model in **3** (default Random Forest)

Step 4: check the predictions save in **5**

Workflow description:

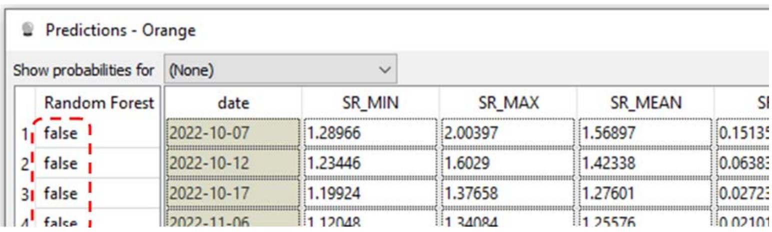
1. Upload of new polygon/s dataset/s in .csv format

```
date,SR_MIN,SR_MAX,SR_MEAN,SR_STD,NDVI_MIN,NDVI_MAX,NI
01/01/2020,2.522996058,4.033557047,3.113798426,0.3680
```

2. Remove Nan values from new dataset (cloud pixel)
3. Load of the selected trained model

   example: RandomForest_Model.pkcls

4. Prediction process

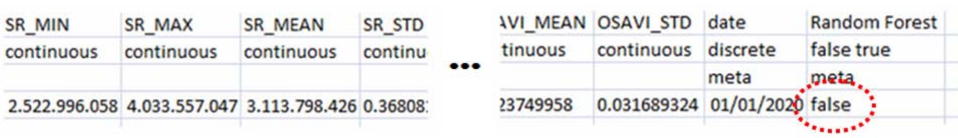


Predictions - Orange

Show probabilities for (None)

	Random Forest	date	SR_MIN	SR_MAX	SR_MEAN	SI
1	false	2022-10-07	1.28966	2.00397	1.56897	0.1513
2	false	2022-10-12	1.23446	1.6029	1.42338	0.0638
3	false	2022-10-17	1.19924	1.37658	1.27601	0.0272
4	false	2022-11-06	1.12048	1.34084	1.25576	0.0210

5. Save prediction in .csv format



SR_MIN	SR_MAX	SR_MEAN	SR_STD	•••	VI_MEAN	OSAVI_STD	date	Random Forest
continuous	continuous	continuous	continu		tinuous	continuous	discrete	false true
							meta	meta
2.522.996.058	4.033.557.047	3.113.798.426	0.36808		23749958	0.031689324	01/01/2020	false