Geo Engine

Geo Engine: Harmonized data access for data analysis pipelines

Dr. Christian Beilschmidt

GWF 2023

Ein Start-up der

Philipps Universität Marburg

Gefördert durch:

Bundesministerium für Wirtschaft und Klimaschutz

aufgrund eines Beschlusses des Deutschen Bundestages

eXIST Existenzgründungen aus der Wissenschaft

ESF Europäischer Sozialfonds für Deutschland

EUROPÄISCHE UNION

Zusammen. Zukunft. Gestalten.

About us

- Geo Engine GmbH
  est. 2021
- Start-up of University of Marburg, Germany
- Geo informatics researchers from biodiversity and remote sensing projects
- EXIST research transfer

Participation in NFDI4Biodiversity

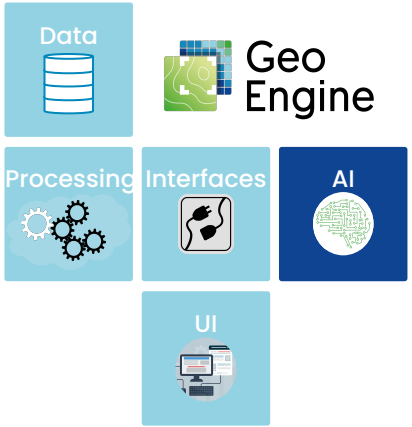
Co-applicant in CropHype (EnMap)

ML training data pipelines (Sentinel 2 aggregates)

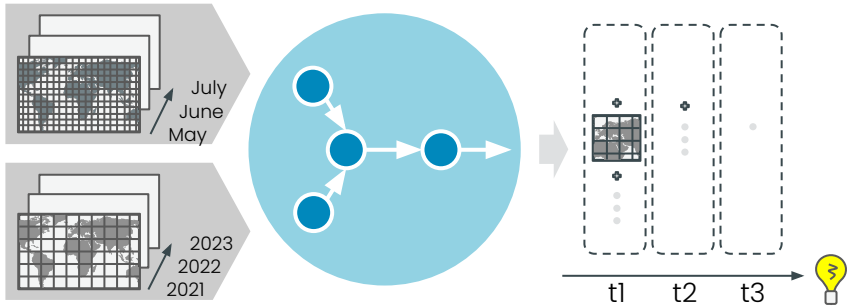
Geo Engine in a nutshell

- Platform for spatio-temporal processing
- Transparent ad-hoc integration of external data via Data Providers
- Process data as time-series instead of files ⇨ Transition from static to temporal analytics

Architecture

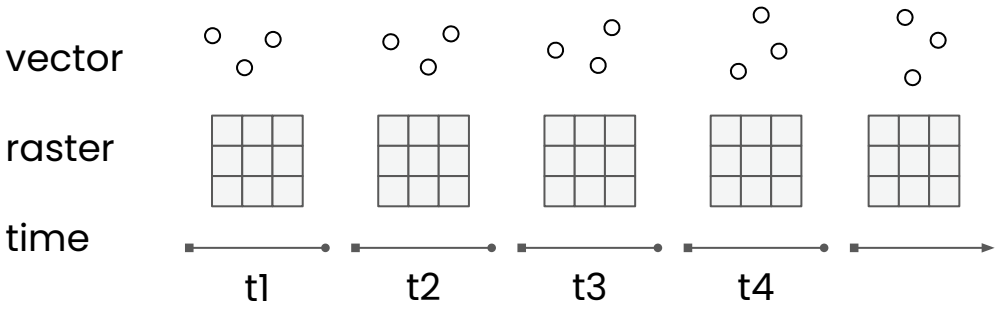


Approach

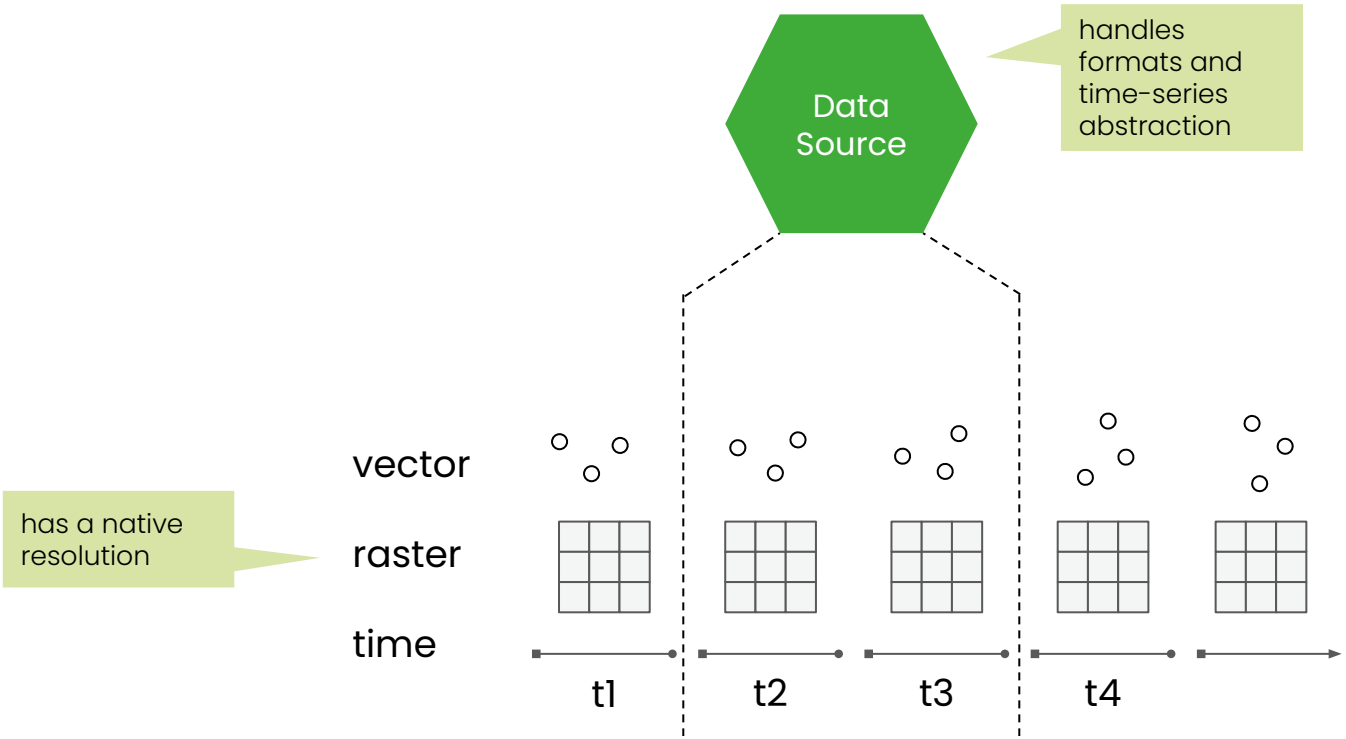


Concept 1: All datasets are time series

has a native resolution



Concept 1: All datasets are time series



Concept 2: Internal and external data

Internal datasets
e.g. point observations

① GeoPackage

② Annotation

Geom	Time	Text	Number
	2023-05-04	This	0.4
	2023-05-04	is	1.3
	2023-05-04	Text	20.4
	...		

③ Integrated time series dataset

External data
e.g. Sentinel-2 data from STAC service

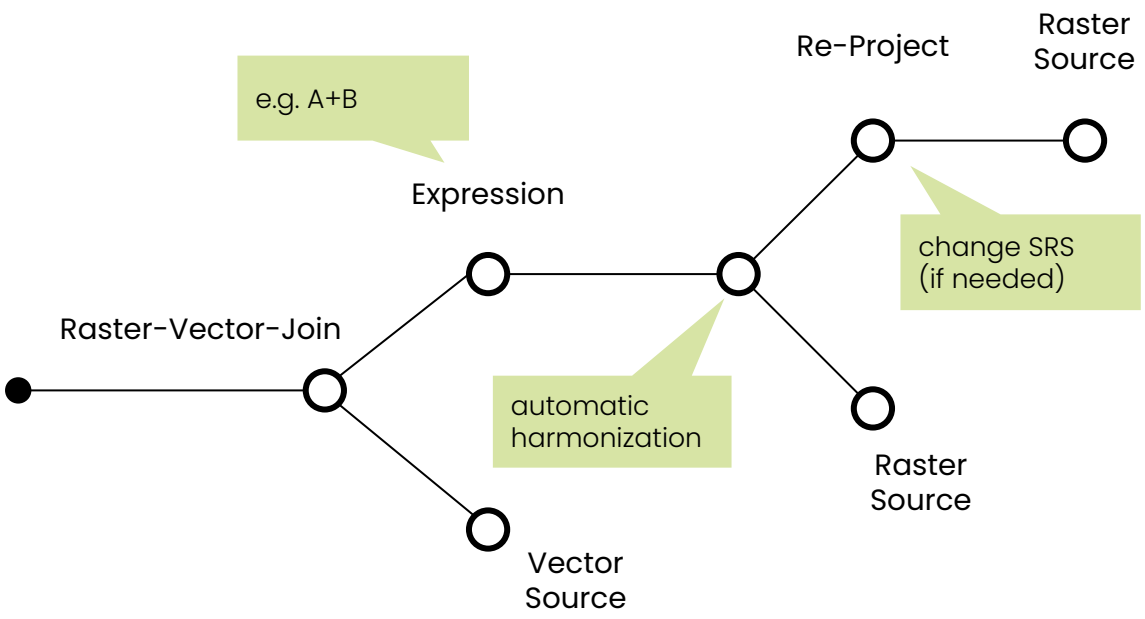
① Sentinel-2 STAC

② Instantiate STAC Provider

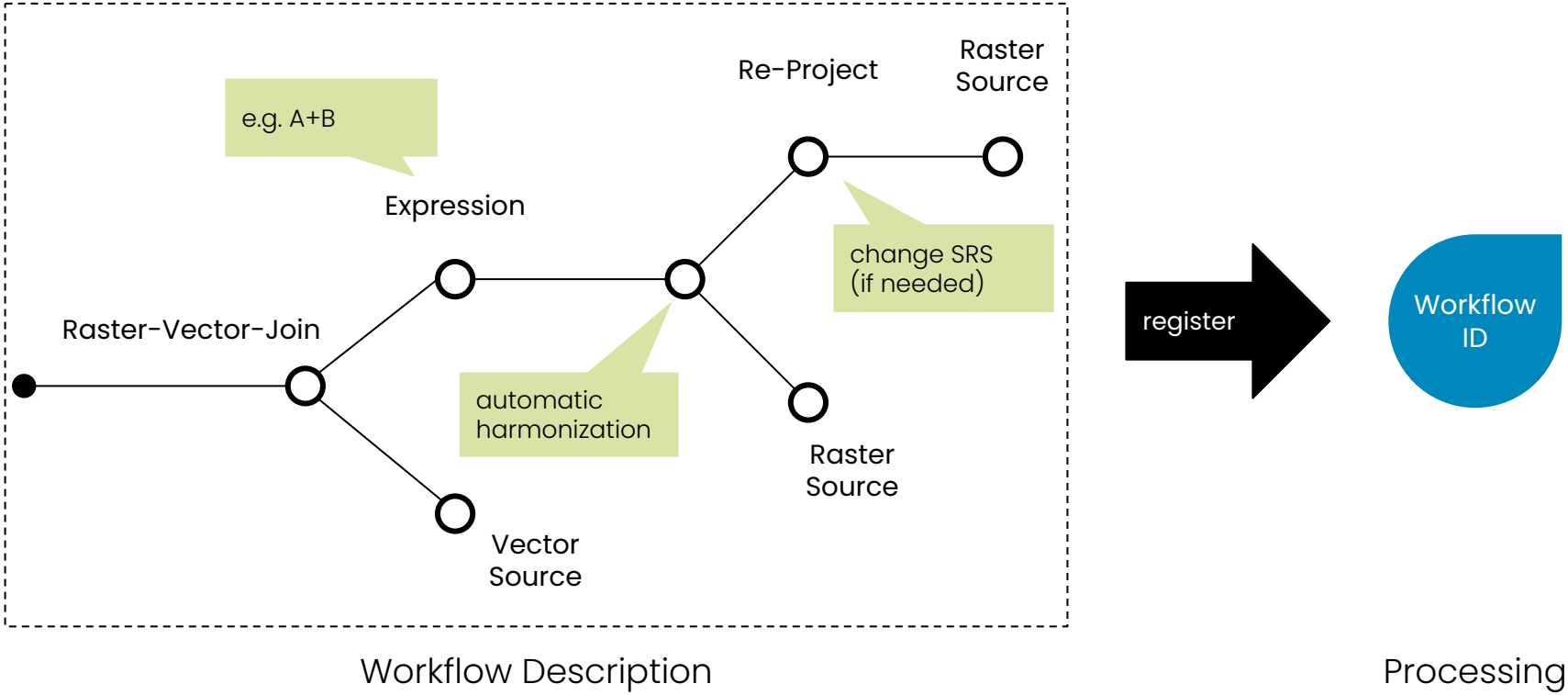
- Implements listing
- Implements resolving datasets

③ Integrated time series dataset

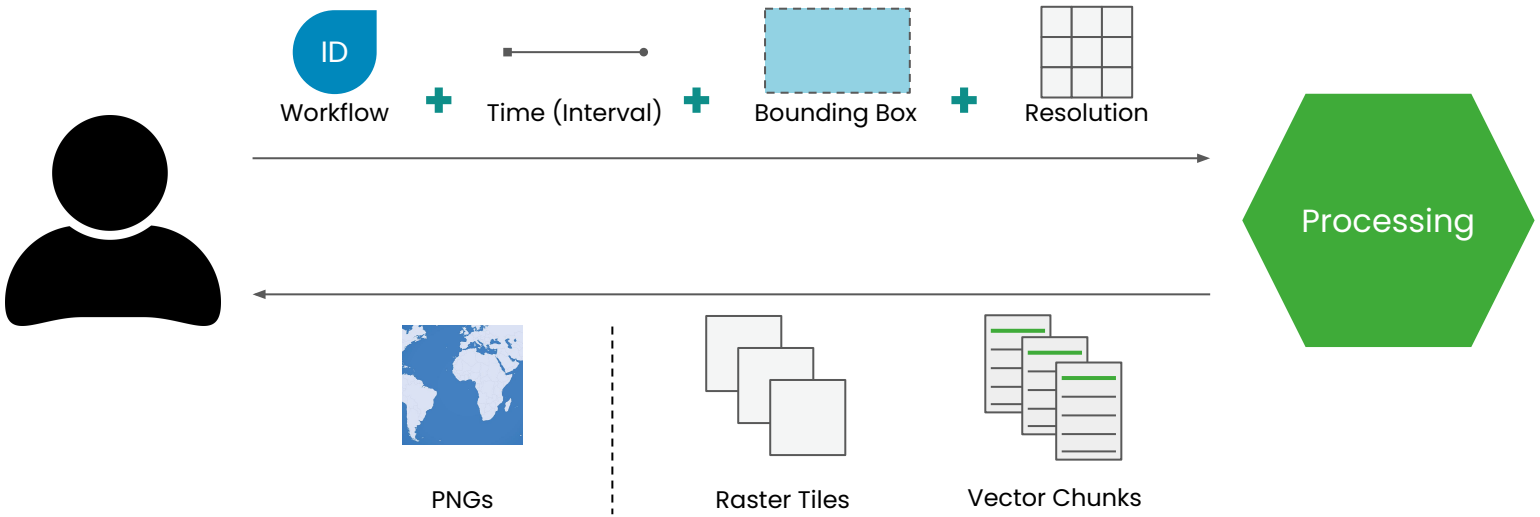
Concept 3: Workflows



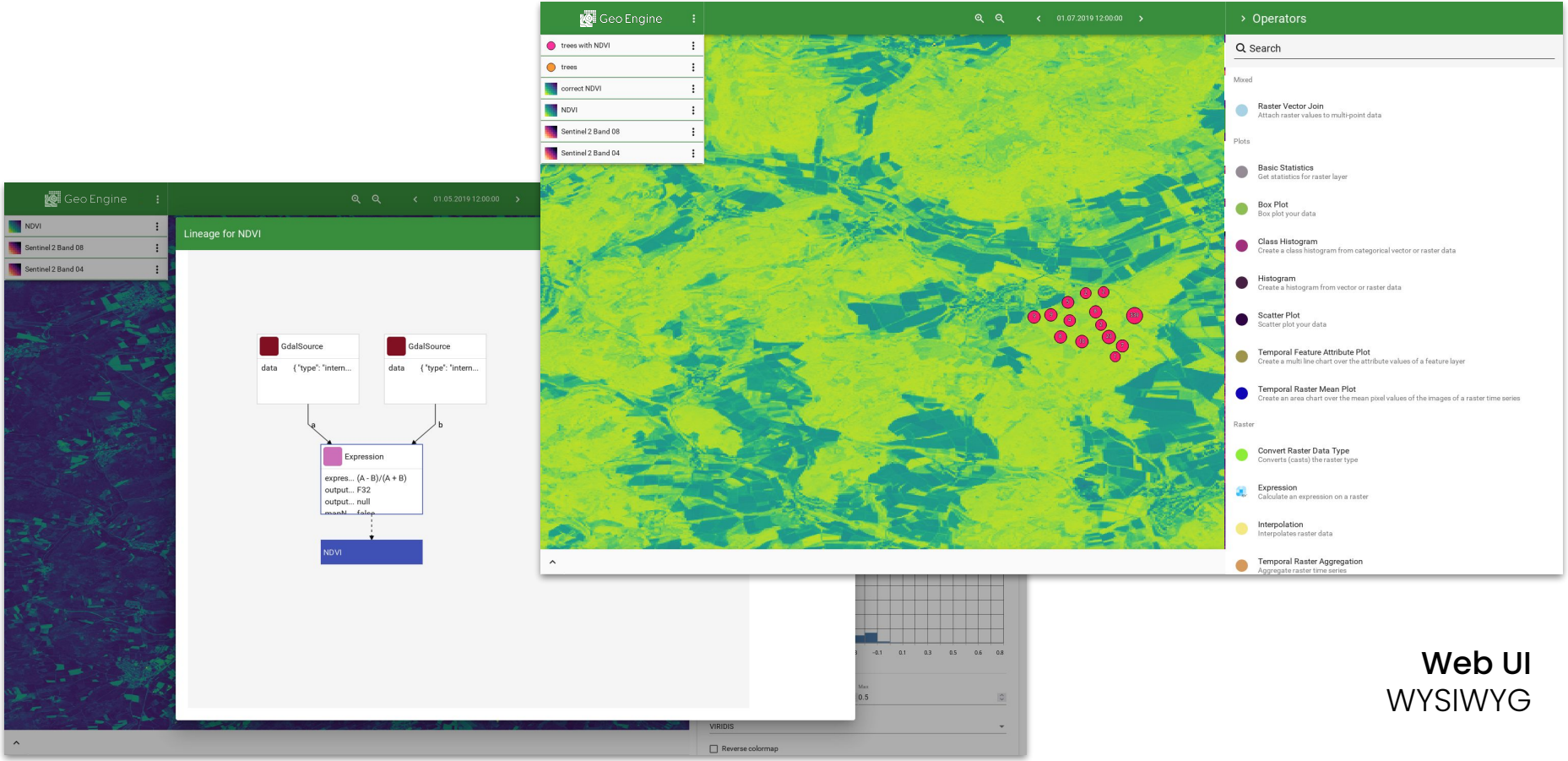
Concept 3: Workflows



Concept 4: Query like a data cube



Concept 5: Access from your favorite environment



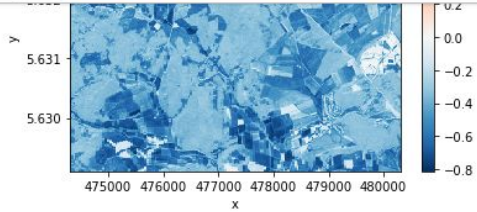
Web UI
WYSIWYG

Concept 5: Access from your favorite environment

File Edit View Run Kernel Tabs Settings Help

Launcher | ndvi_inverse_points.ipynb

Code



Extend the workflow

[14]:
```
workflow_spec = {
    "type": "Raster",
    "operator": {
        "type": "Expression",
        "params": {
            "expression": "(-1.0* A)", # this is the NDVI formula
            "outputType": "F32",
            "mapNoData": False
        },
        "sources": {
            "a": ndvi_workflow.workflow_definition()["operator"]
        }
    }
}
inverse_workflow = ge.register_workflow(workflow_spec)
inverse_workflow
```

[14]: ee9eb3b4-12a8-511b-b11b-6d40dff80efb

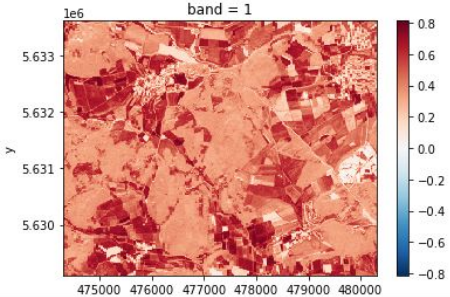
Request inverse data

[15]:
```
inverse_array = inverse_workflow.get_xarray(query_rectangle)
```

Query the NDVI raster

[16]:
```
inverse_array.plot()
```

[16]: <matplotlib.collections.QuadMesh at 0x7f9e1625da20>

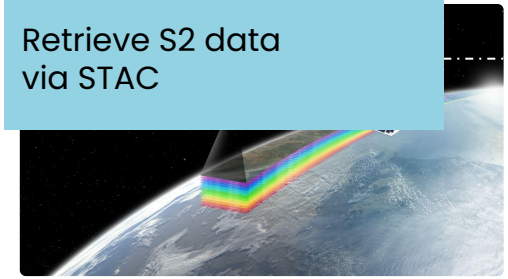


Simple 0 6 Python 3 (ipykernel) | Idle

Jupyter Notebooks
Programmatically

Example: Random Forest using custom point and Sentinel data

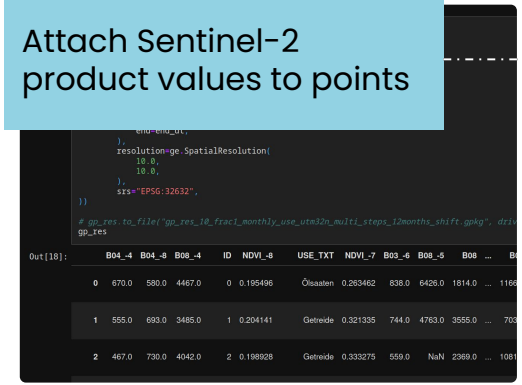
Retrieve S2 data via STAC



Monthly mean of cloud-free NDVI

Load sensor training points

Attach Sentinel-2 product values to points



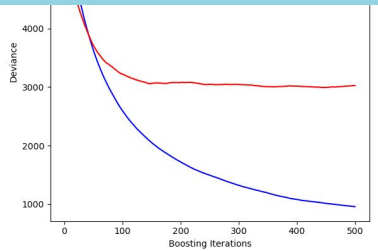
```
        end=end_dt,
        ),
        resolution=ge.SpatialResolution(
            10.0,
            10.0,
        ),
        srs="EPSG:32632",
    })

# gp_res.to_file("gp_res_10_frac1_monthly_use_utm32n_multi_steps_12months_shift.gpkg", dri
gp_res
```

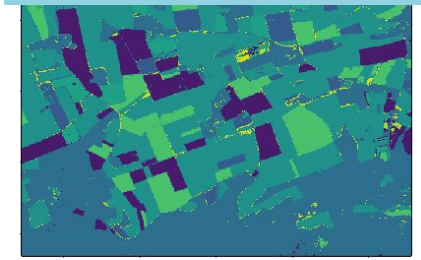
Out[18]:

	B04_-4	B04_-8	B08_-4	ID	NDVI_-8	USE_TXT	NDVI_-7	B03_-6	B08_-5	B08	...	B
0	670.0	580.0	4467.0	0	0.195496	Ölsaaten	0.263462	838.0	6426.0	1814.0	...	1166
1	555.0	693.0	3485.0	1	0.204141	Getreide	0.321335	744.0	4763.0	3555.0	...	703
2	467.0	730.0	4042.0	2	0.198928	Getreide	0.333275	559.0	NaN	2369.0	...	1081

Stream data in RF model using Python & Jupyter



Apply to all pixel values



Summary

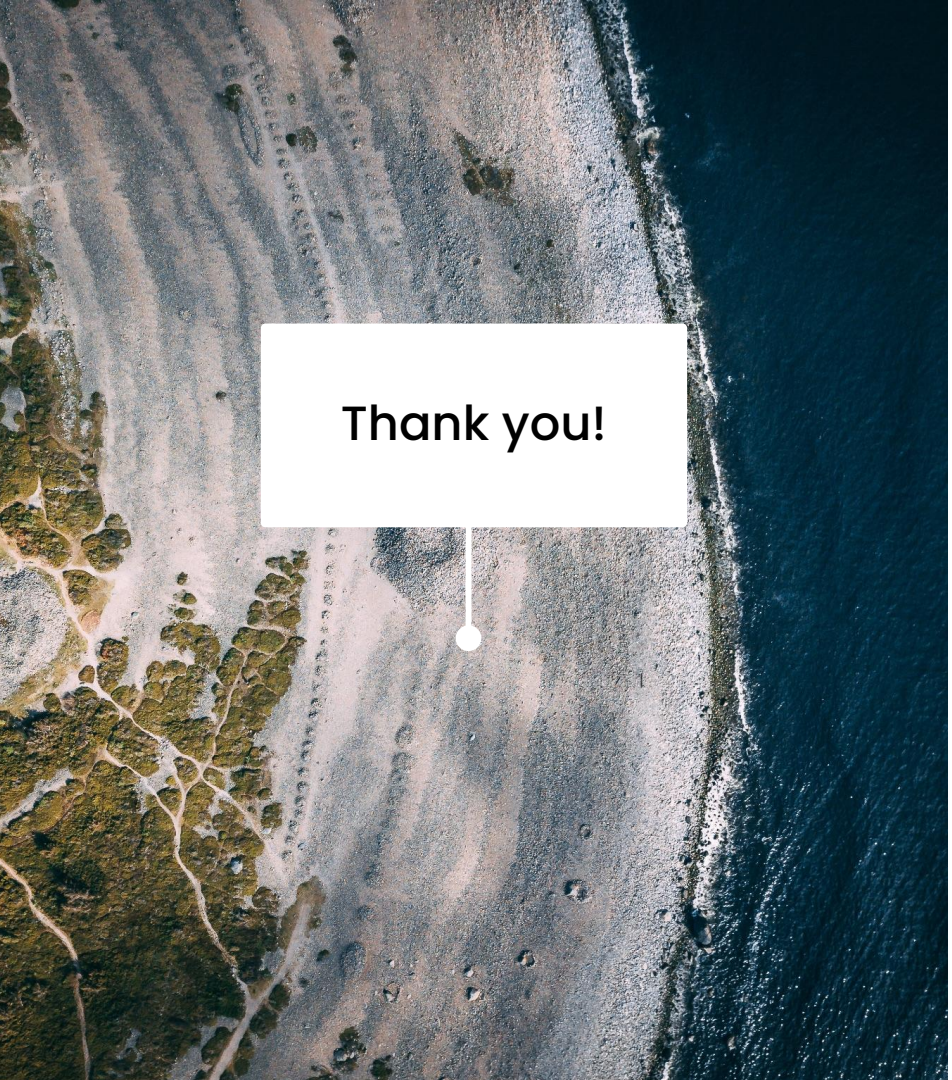
- Geo Engine platform for geo processing & analysis
- Native time series
- Worklows
- Data harmonization
- Low-code
- Jupyter notebooks

Outlook

- Integrate machine learning framework(s) into Geo Engine
- ML ingestion inside
- ML as workflow operator

Call for Cooperation

- Data ingestion use case
- Data products from different EO sources
- Proof-of-concepts



Thank you!

Geo Engine

Geo Engine GmbH
Am Kornacker 68
35041 Marburg

info@geoengine.de

geoengine-de

@teamgeoengine

www.geoengine.de