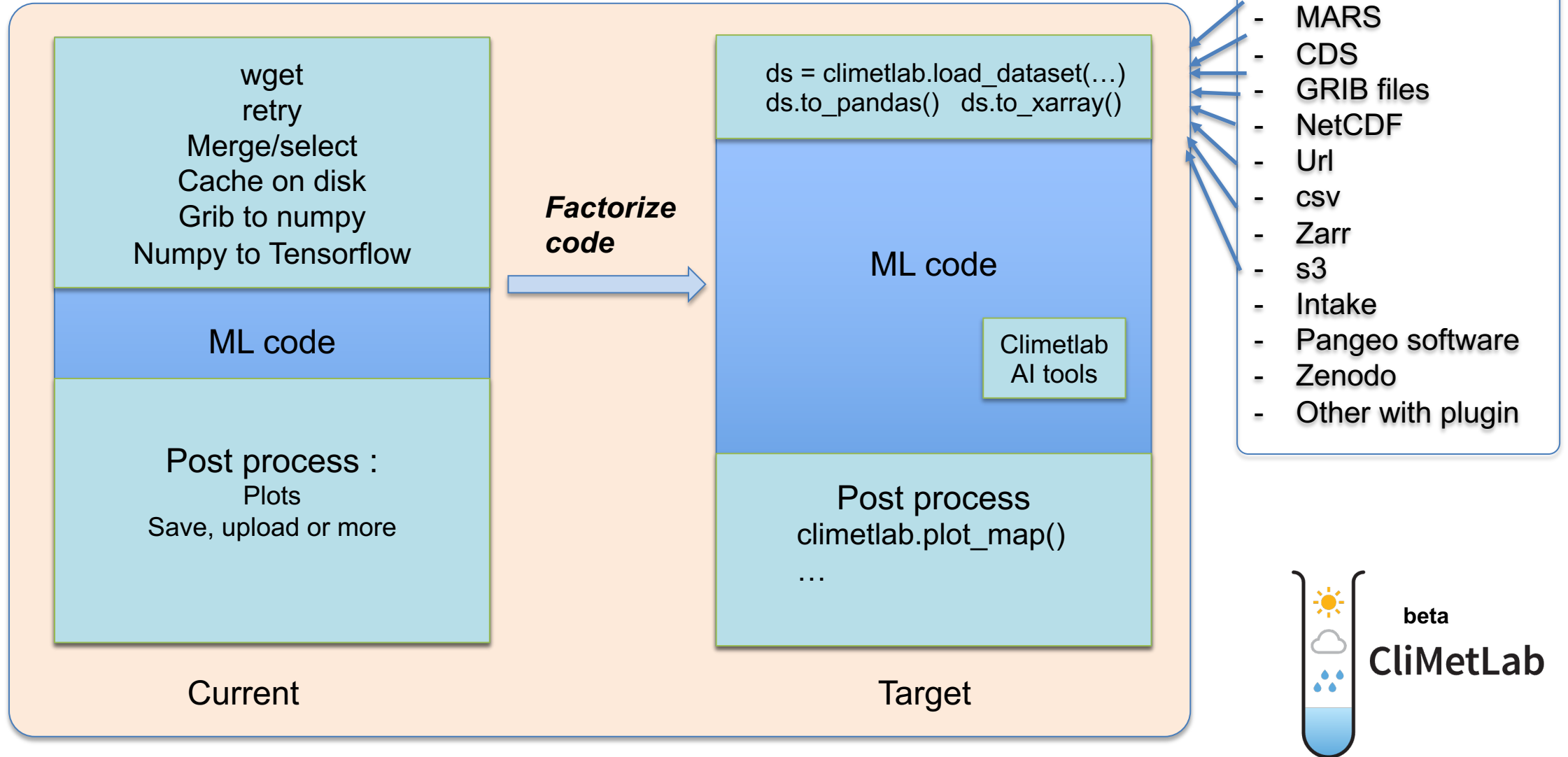


CliMetLab and Pangeo use case: Machine learning data pipeline for sub-seasonal to seasonal prediction (S2S)

Florian Pinault, Aaron Spring, Frederic Vitart, and Baudouin Raoult

Looking at some Jupyter Notebooks



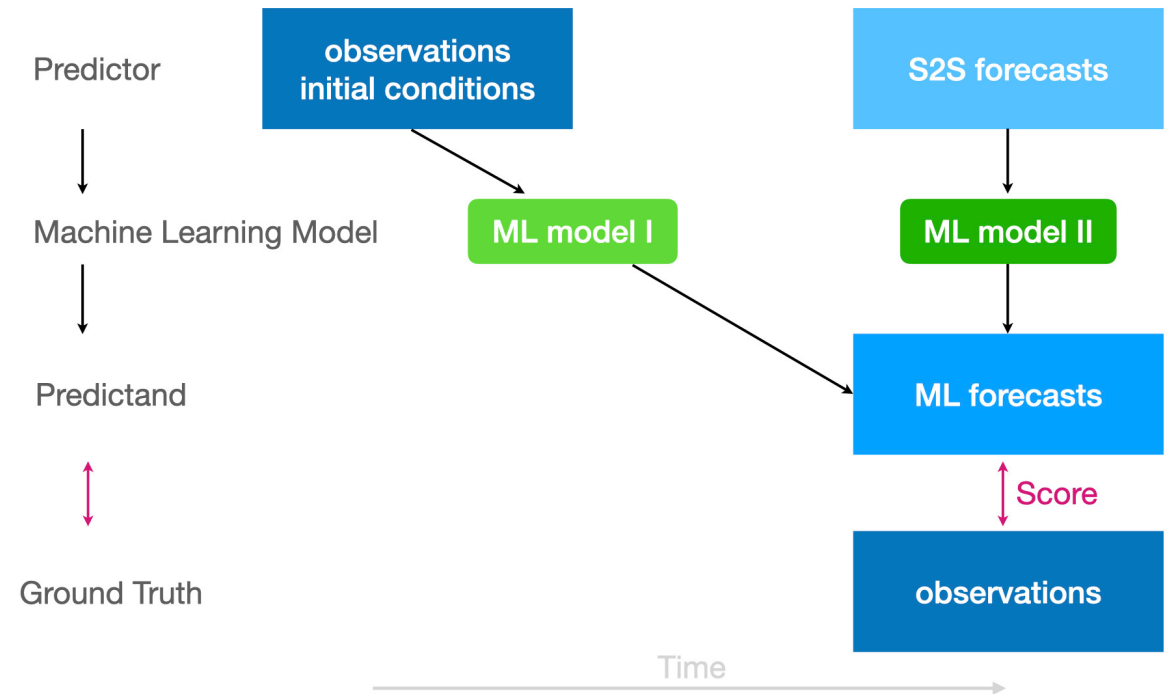
Use case: provide data for the S2S AI challenge.

Data provided for the challenge

- Models (input)
 - ECMWF, NCEP, ECCC only.
 - Daily forecast up to 46 days.
 - Reforecast between 1999 to 2019.
 - Using 2020 as test data for evaluation.
 - 20 parameters (t2m, tp, u, v, gh,)
 - Resolution 1.5 degree, regular grid.
- Observations (output)
 - Targets: t2m and tp terciles (from IRIDL).
 - Several grids available.
 - Biweekly averaged.
- Benchmark
 - Probabilistic ECMWF recalibrated.

Machine Learning tasks

- I) ML Forecast directly from observations (replace classical models)
- II) ML Forecast using S2S forecast (improve classical models)



Data: multiple formats (incl. ZARR).

Multiple formats

- GRIB: from the ECMWF archive database (MARS)
- NetCDF: converted from GRIB.
- ZARR: converted from netCDF.

Hosted on the European Weather Cloud (EWC)

- S3 bucket
- Joined initiative ECMWF-EUMETSAT

ZARR data.

- Generated using a Pangeo recipe
- Needed to make layout choices (chunks).
 - Best choice depends on the use case
- “Move the data to the code” vs “move the code to the data”.
 - Paradigm shift.
 - Inside the EWC cloud, access to ZARR data is fast.
 - Data users outside of the EWC cloud prefer GRIB/NetCDF.

Data: consistent API for all dataset in the S2S AI challenge.

Multiple formats

- GRIB
- NetCDF
- ZARR

Multiple datasets

- Training-input: hindcast data (< 2020)
- Test-input: 2020 forecast data.
- Training and test output: observations
- Benchmark: recalibrated data.

Multiple naming conventions

- "t2m" or "2t"
- "2020-01-02" or datetime(2020,1,02)
- "hindcast-input" or "training-input"

Climetlab provides a consistent interface (API). Using a python plugin linking also to documentation and additional code.

```
d = climetlab.load_dataset("s2s-ai-challenge-forecast-input", parameter="t2m", date="20200102")
d = climetlab.load_dataset("s2s-ai-challenge-forecast-input", parameter="t2m", date="20200102", format="netcdf")
d = climetlab.load_dataset("s2s-ai-challenge-forecast-output", parameter="t2m", date="20200102")
```

- Download, cache data, authenticate, retry, etc.
- Provides familiar python objects (xarray, pandas, numpy, tensorflow).

```
d.to_xarray() -> xarray dataset
d.to_numpy() -> numpy array
d.to_pandas() -> pandas dataframe
```

“Give me my data”: a consistent interface (API).

```
climetlab.load_dataset("dataset-name", keyword="1", keyword2=2)
```

Consistent interface (API) for the S2S AI challenge.

```
d = climetlab.load_dataset("s2s-ai-challenge-forecast-input", parameter="2t", date="2020-01-02", format="grib")  
d = climetlab.load_dataset("s2s-ai-challenge-forecast-output", parameter="t2m", date="20200102")
```

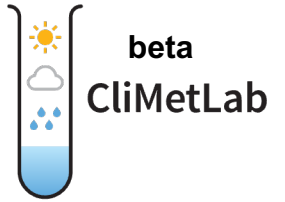
Consistent interface (API) with other datasets and source of data.

```
d = climetlab.load_source("cds", param="t2m", ...) [Climate Data Store]  
d = climetlab.load_source("eumetsat", ...) [Collaboration EUMETSAT in progress]  
d = climetlab.load_source("mars", param="t2m", ...) [MARS archive]  
d = climetlab.load_source("ecmwf-open-data", param="t2m", ...)  
d = climetlab.load_dataset("maelstrom-radiation-radiation", year="2021", ...)
```

```
d.to_xarray() -> xarray dataset  
d.to_numpy() -> numpy array  
d.to_pandas() -> pandas dataframe
```

- Download, cache data, authenticate, retry, etc.
- Provides familiar python objects (xarray, pandas, numpy, tensorflow).

Conclusion



```
climetlab.load_dataset("dataset-name", keyword="1", keyword2=2)
```

Open source on GitHub ! (<https://github.com/ecmwf/climetlab>)

Beta version ! Some parts already usable (and used)

We will need you !

- Create your datasets plugins
- Try it and add feature requests on github
- Provide solutions and use cases
- Report bugs on github and add pull request to fix them
- Communicate through github issues