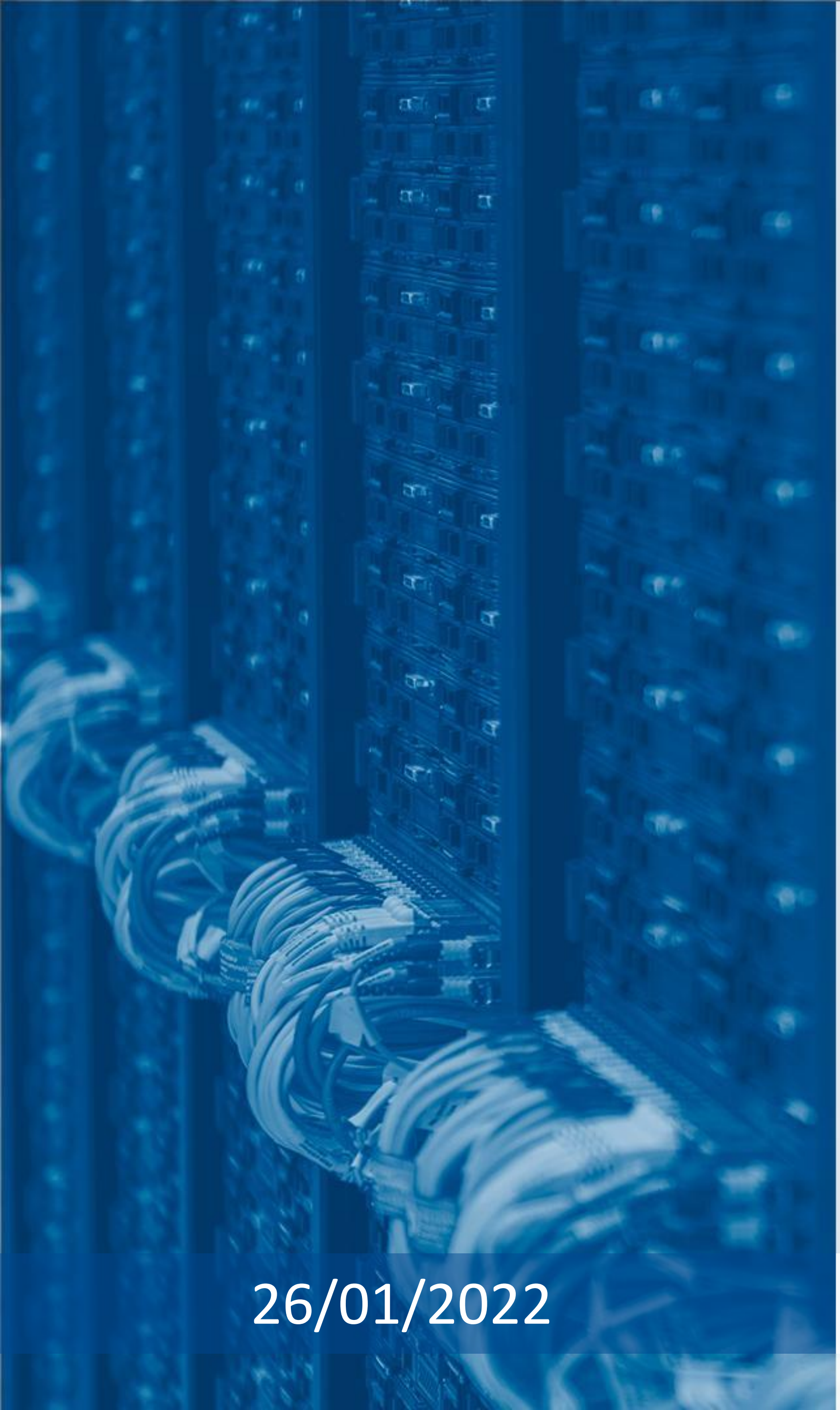


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

BSC contribution to the WMO-S2S-AI CHALLENGE

Llorenç Lledó,

Sergi Bech, Lluís Palma, Andrea Manrique & Carlos Gómez

26/01/2022

S2S monthly webinar

The BSC team

ANDREA MANRIQUE
Sub-seasonal prediction services

SERGI BECH
Maths & computer science student

LLORENÇ LLEDÓ
Statistical post-processing of climate predictions

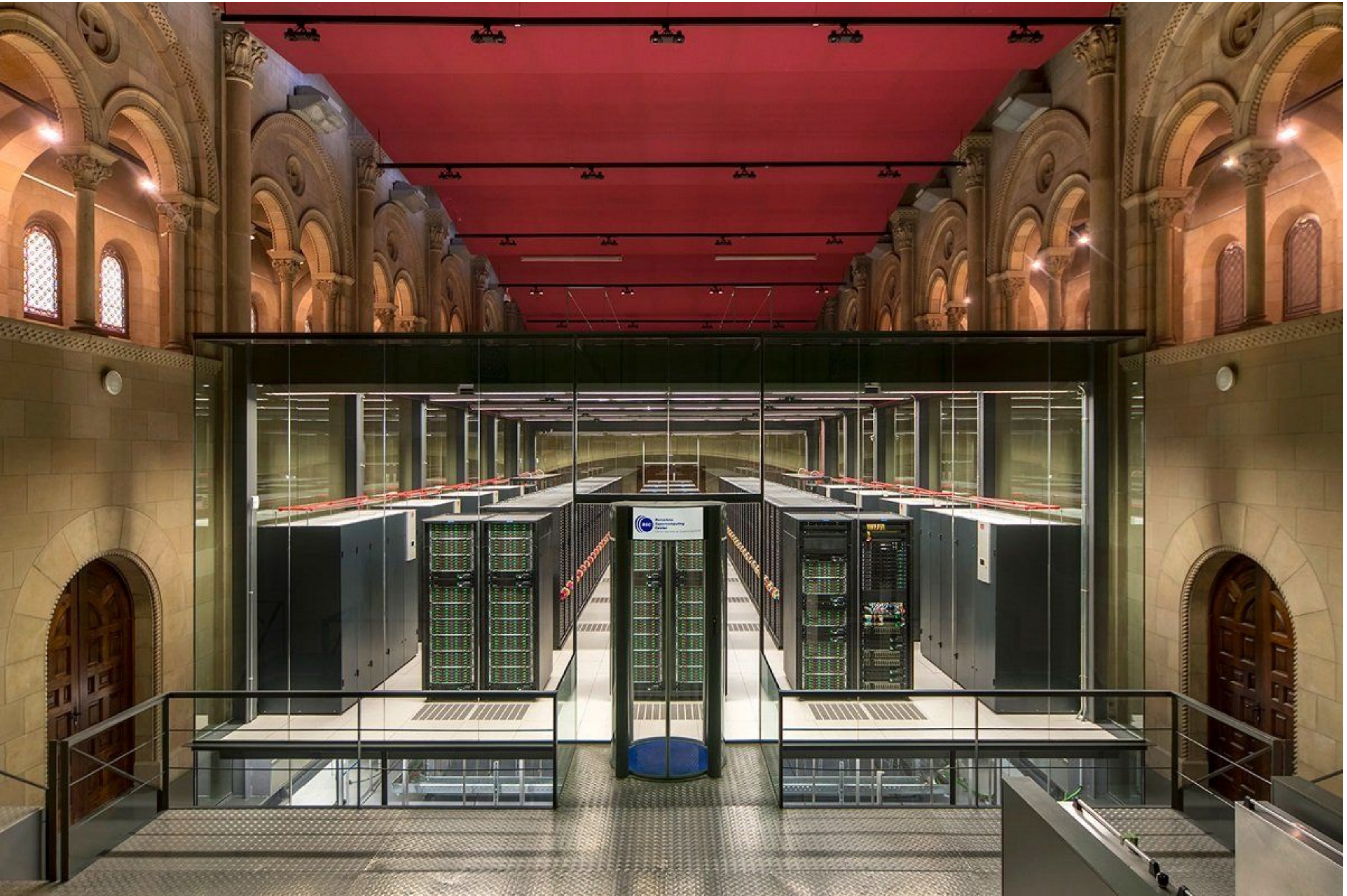
CARLOS GOMEZ
AI for Earth Science problems

LLUÍS PALMA
Operational S2S predictions engineer

BSC Barcelona Supercomputing Center
Centro Nacional de Supercomputación

MareNostrum 4

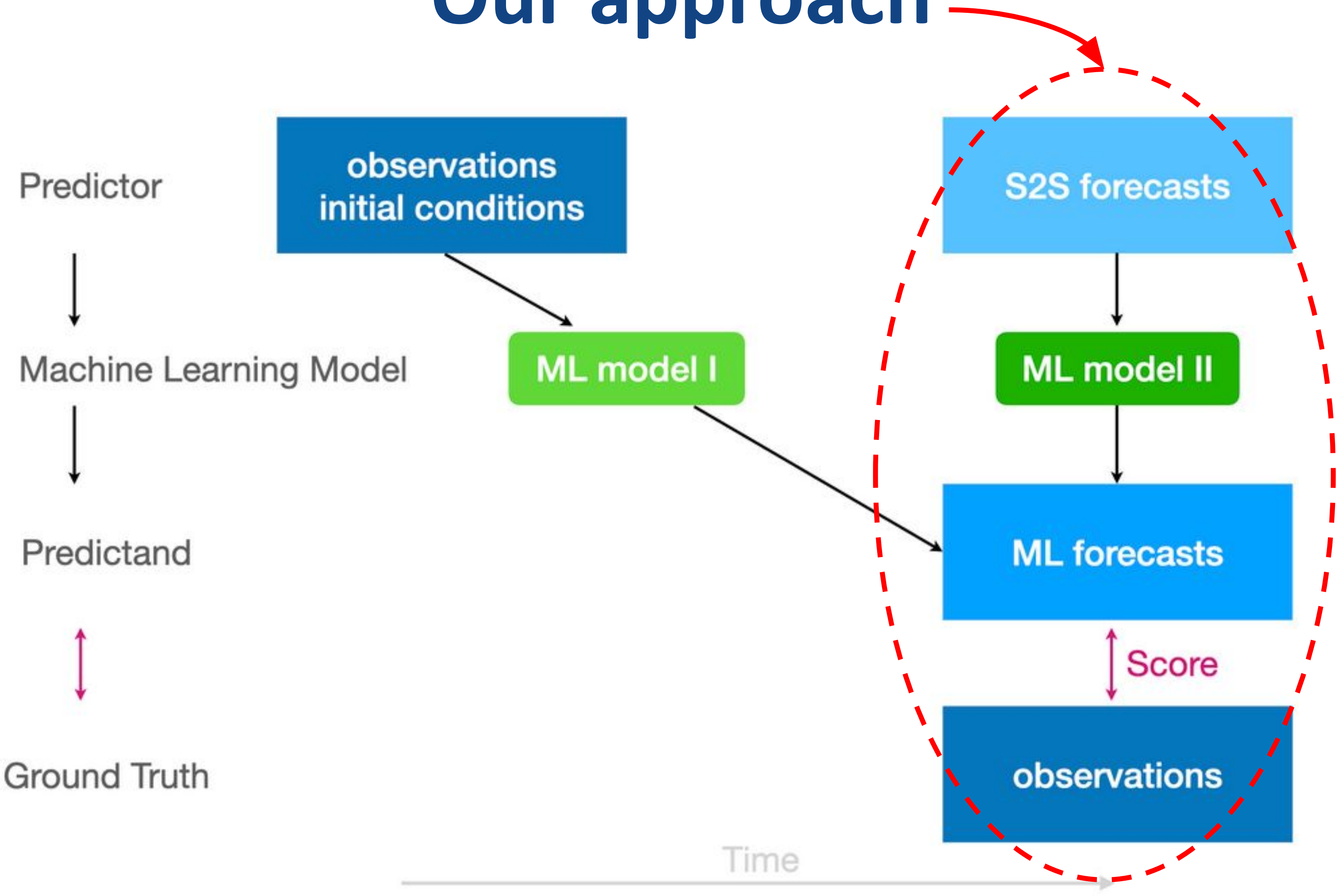
“the most beautiful data center in the world”



SUPERCOMPUTING RESOURCES

Platform:	CTE-Power9
Processors requested:	120 CPUs
Available memory:	~ 500 Gb
Consumed memory:	~ 100 Gb

Our approach



Our approach

“A point by point statistical correction of ECMWF forecasts that transforms raw ensemble output into calibrated tercile probabilities”

- a model is trained separately for each grid point, variable and lead time
- the predictors are ECMWF forecasts for the same variable
- we train 4 methods and pick the best at each location
- we do not need a post-processed ensemble, just the tercile probabilities

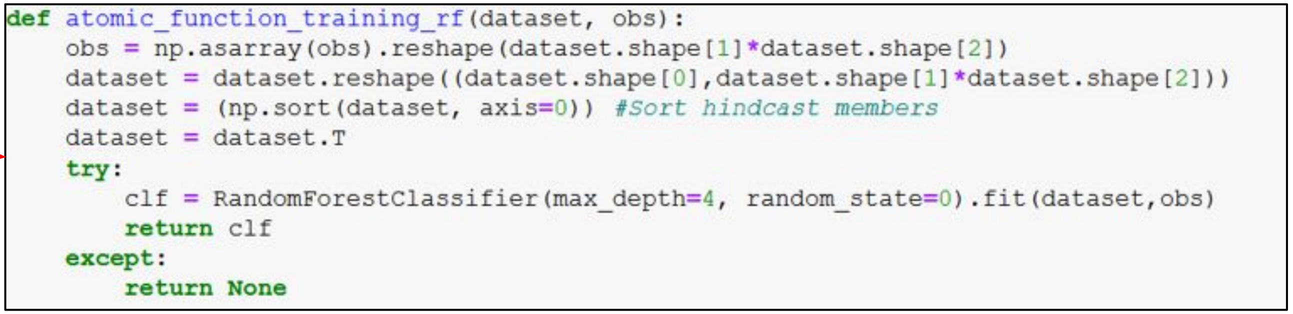
Implementation

A ***Jupyter*** notebook written in ***Python*** with

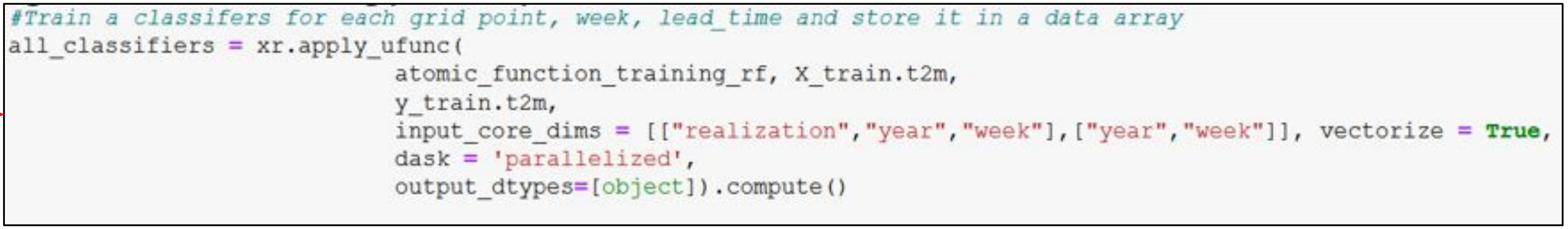
- ***xarray*** for loading/working with multidimensional data
- ***dask*** for parallelizing computations
- ***scikit-learn*** to implement ML techniques

`atomic function`
trains one method for one grid point, lead-time and variable.

`apply_ufunc`
calls the `atomic_function`for each grid point, lead-time and variable and saves output as `xarray`.



```
def atomic_function_training_rf(dataset, obs):
    obs = np.asarray(obs).reshape(dataset.shape[1]*dataset.shape[2])
    dataset = dataset.reshape((dataset.shape[0],dataset.shape[1]*dataset.shape[2]))
    dataset = (np.sort(dataset, axis=0)) #Sort hindcast members
    dataset = dataset.T
    try:
        clf = RandomForestClassifier(max_depth=4, random_state=0).fit(dataset,obs)
        return clf
    except:
        return None
```



```
#Train a classifers for each grid point, week, lead_time and store it in a data array
all_classifiers = xr.apply_ufunc(
                             atomic_function_training_rf, X_train.t2m,
                             y_train.t2m,
                             input_core_dims = [["realization","year","week"],["year","week"]], vectorize = True,
                             dask = 'parallelized',
                             output_dtypes=[object]).compute()
```

Datasets

For **training**:

- ECMWF hindcasts of *t2m* and *tp* for 2000-2019, 11 members
- CPC categorical observations for 2000-2019

For **prediction**:

- ECMWF forecasts of *t2m* and *tp* for 2020, 51 members

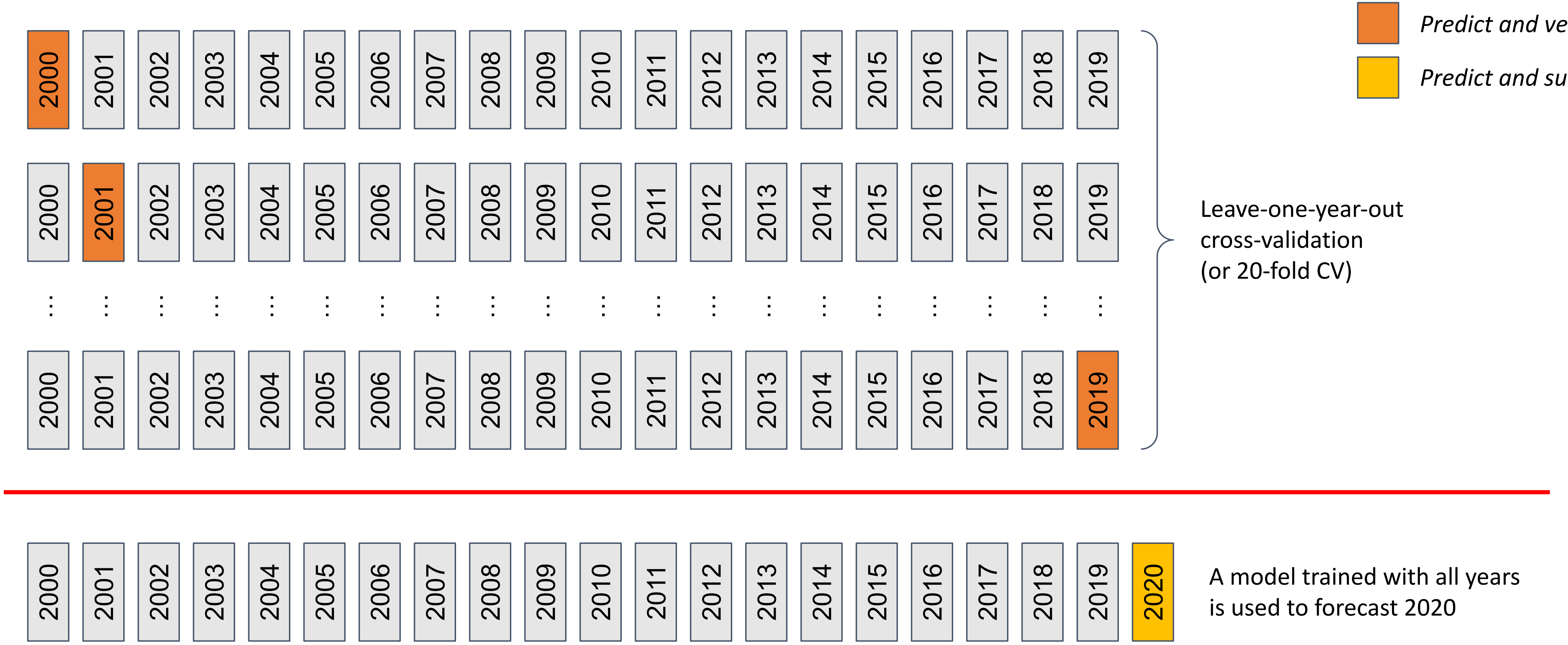
(all data bi-weekly aggregated and as
provided by the challenge)

train/test/predict strategy

⬜ *Train*

🟧 *Predict and verify*

🟨 *Predict and submit*



🟧 2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
2000	🟧 2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	🟧 2019

Leave-one-year-out
cross-validation
(or 20-fold CV)

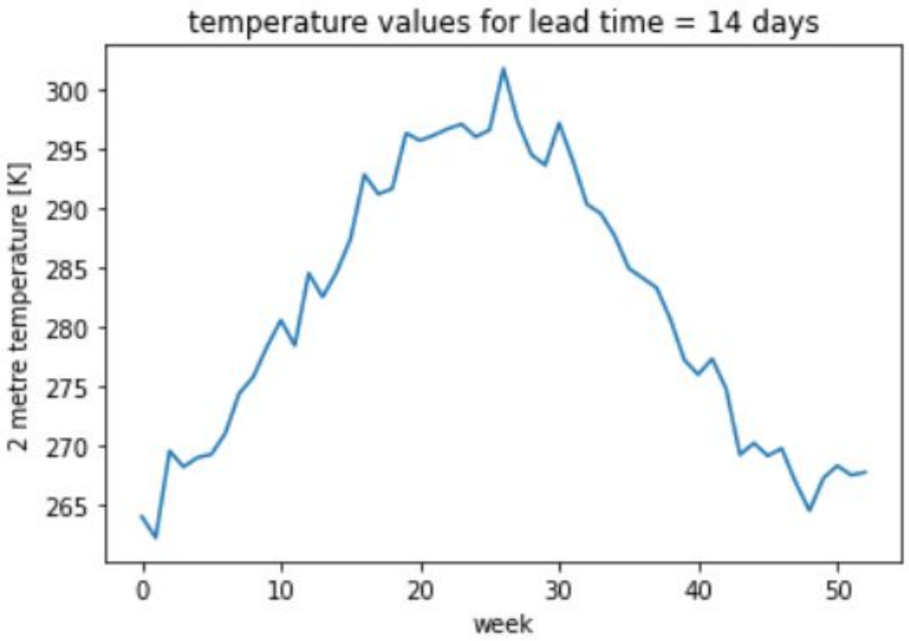
2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	🟨 2020

A model trained with all years
is used to forecast 2020

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

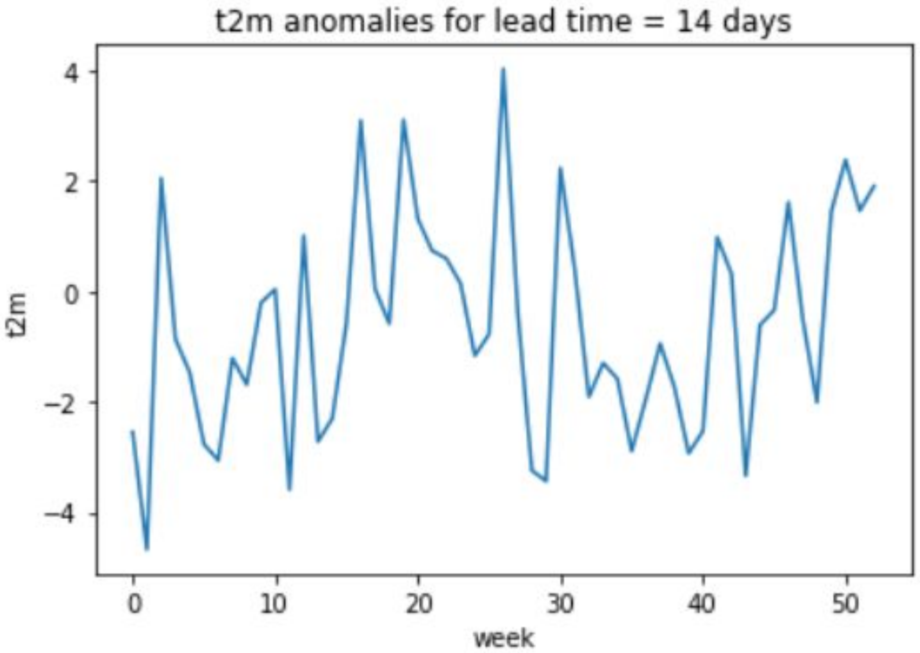
The performance for 2020 can be thought of as a random draw from the 20-fold CV. We use the median RPSS across years to compare methods.

Data pre-processing



Large **sample sizes** are required for ML methods

- Compute anomalies wrt hindcast climatology
- Train all weeks of the year together



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

A hierarchy of methods

	Climatology	Raw ECMWF	Logistic Regression	Random Forest
Predictors	None	Full ensemble	Ens. mean	Full ensemble
Training parameters	None	2 thresholds per week and grid point	2 coefficients per grid point	8 per grid point
Hyperparameters	None	None	None	2 (fixed)
Training samples	None	20 years per week	20y*53w = 1060	20y*53w = 1060
Features	None	11/51	1	11/51
Predictors as anomalies	-	no	yes	yes
num. models	1	53 weeks * 2 vars * 2 leadtimes * 29040 grid points	2 vars * 2 leadtimes * 29040 grid points	2 vars * 2 leadtimes * 29040 grid points

← simple methods

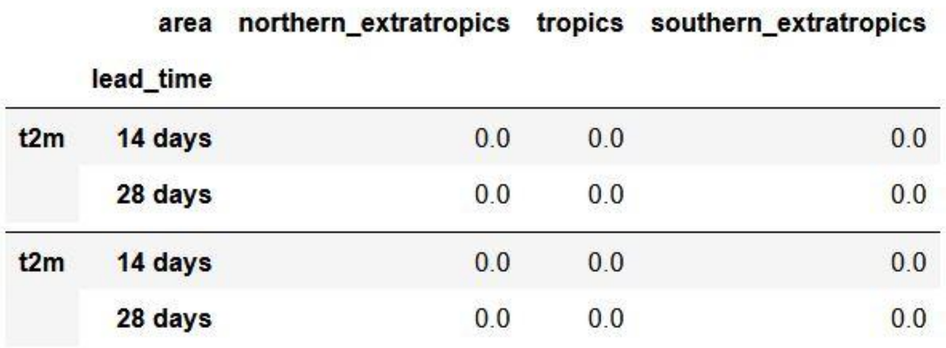
complex methods →

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

Method 1 - Climatology

METHOD
Probability of ⅓ of observing each tercile category

RATIONALE
Ensure we don’t perform worse than climatology



	area	northern_extratropics	tropics	southern_extratropics
	lead_time			
t2m	14 days	0.0	0.0	0.0
	28 days	0.0	0.0	0.0
t2m	14 days	0.0	0.0	0.0
	28 days	0.0	0.0	0.0

RPSS 2000-2019

Method 2 - Raw ECMWF forecasts

METHOD

Count proportion of ECMWF members exceeding the tercile edges

RATIONALE

Ensure we don’t perform worse than ECMWF raw forecasts

TRAINING

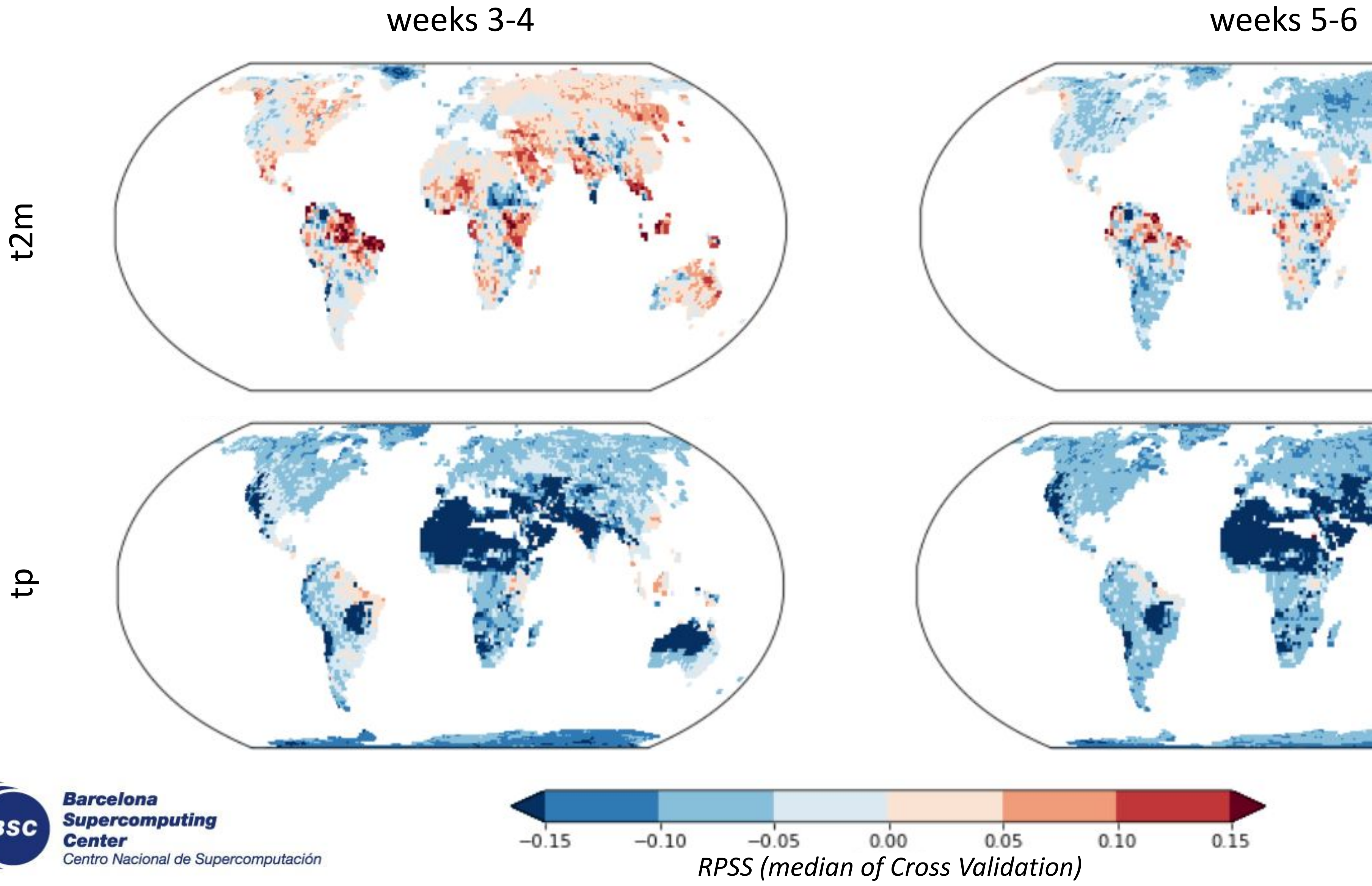
For each week of the year compute the tercile edges in the hindcast (2000-2019)

Implicit bias adjustment

PREDICTORS

All ECMWF ensemble members of the variable of interest

Raw ECMWF: RPSS 2000-2019

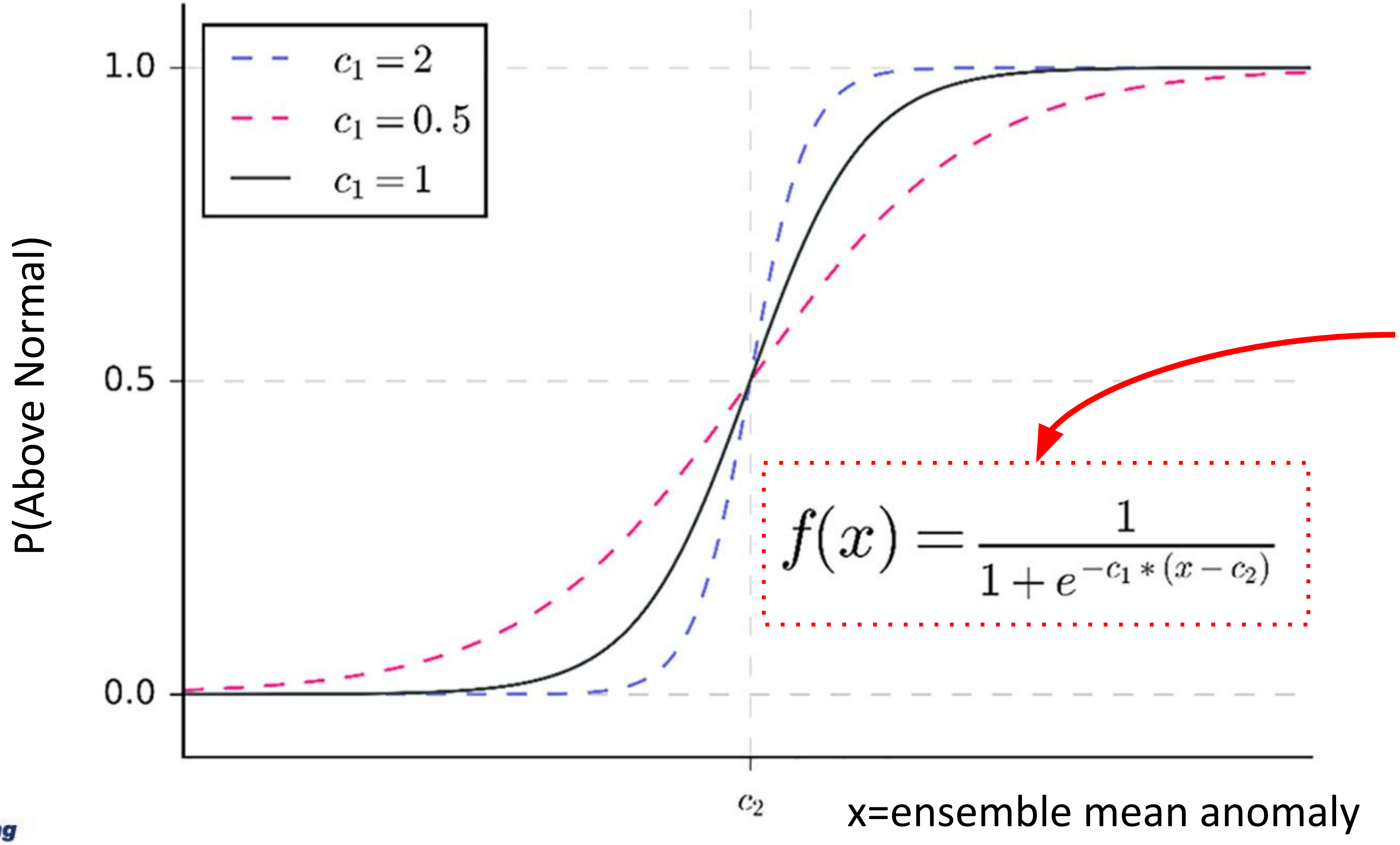


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Method 3 - Logistic regression

RATIONALE

The higher the ensemble mean, the highest the probabilities of above normal conditions

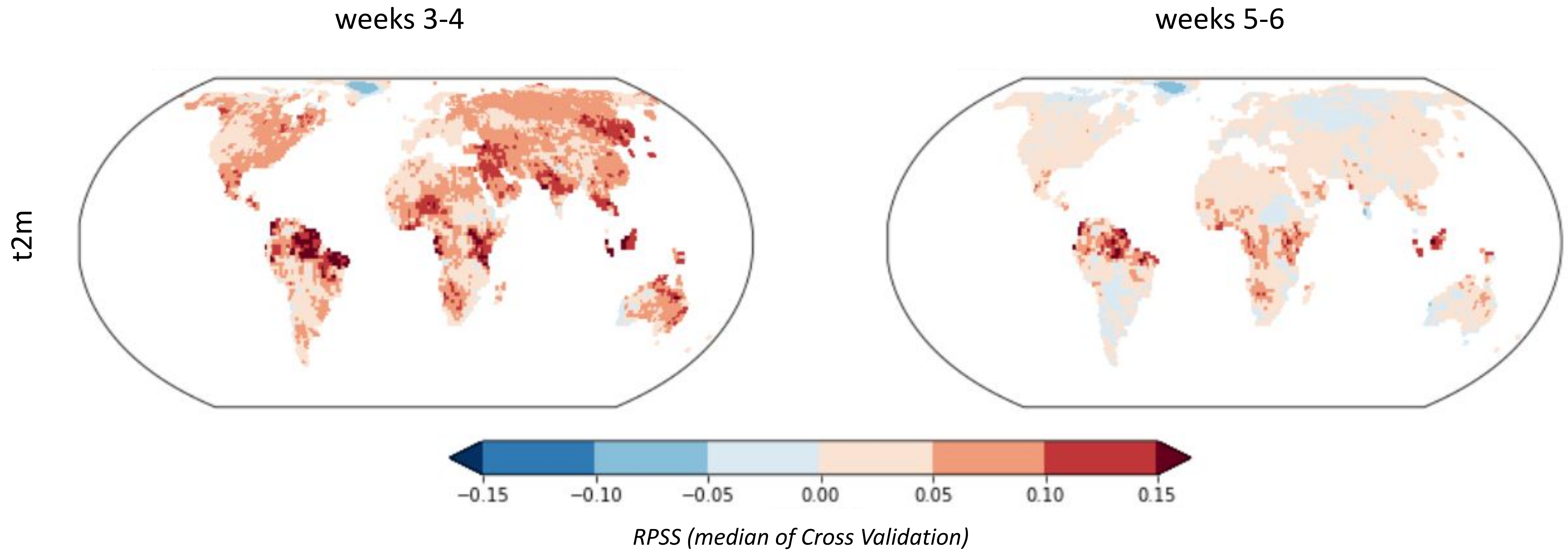


$$f(x) = \frac{1}{1 + e^{-c_1 * (x - c_2)}}$$

the sigmoid (or logistic) function

adapted from Leibovich-Raveh et al. (2018)

Logistic Regression: RPSS 2000-2019



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Method 4 - Random Forest

RATIONALE

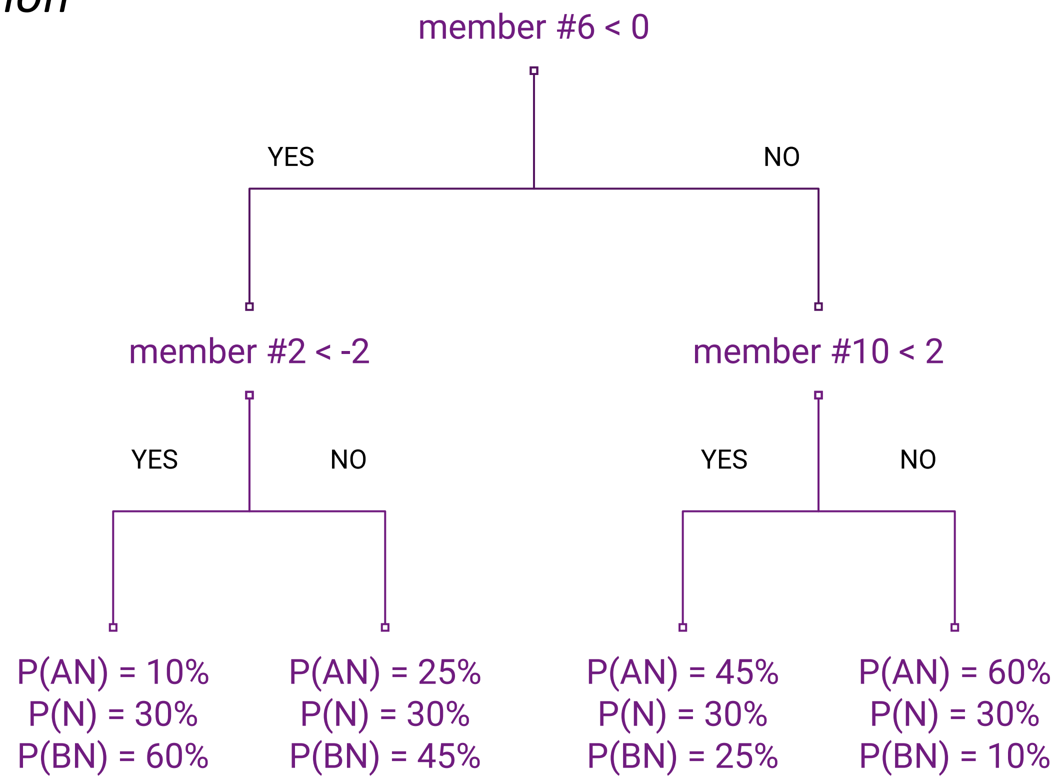
Employ information from all the ensemble distribution

TRICKS

- *Sort ensemble members before train/predict*
- *Subset members 1,6,11,...,51 for prediction*

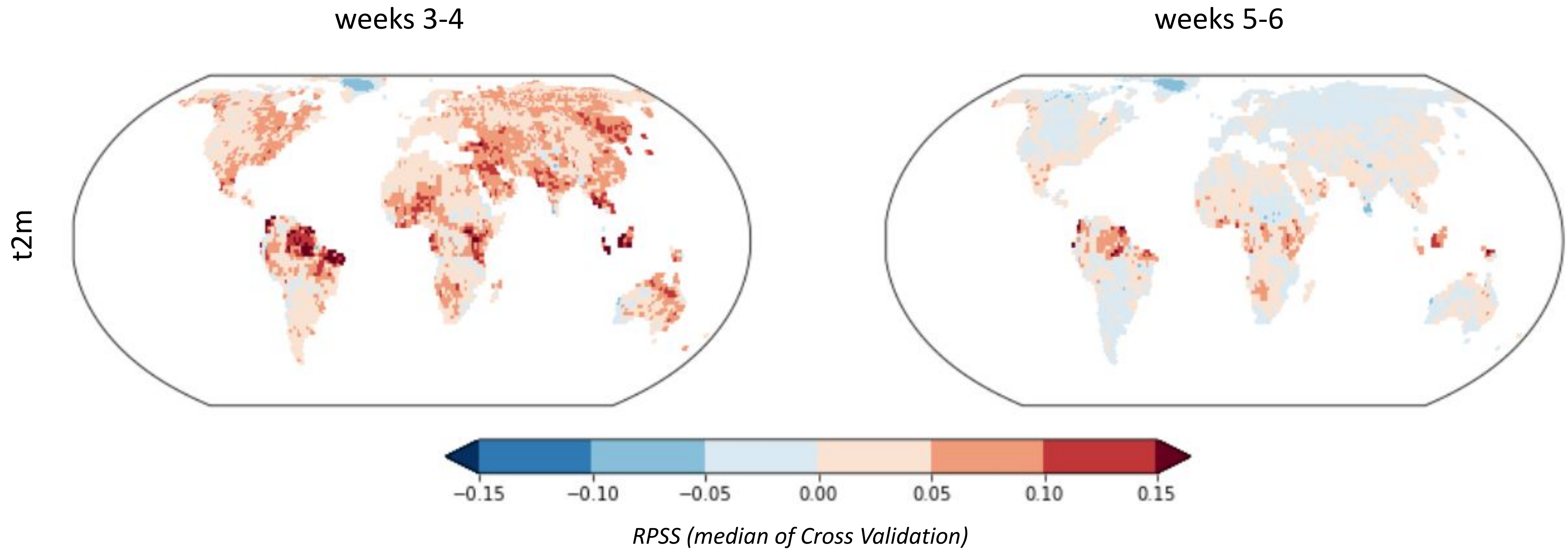
HYPERPARAMETERS (fixed)

- *Depth = 4*
- *Number of trees = 100*



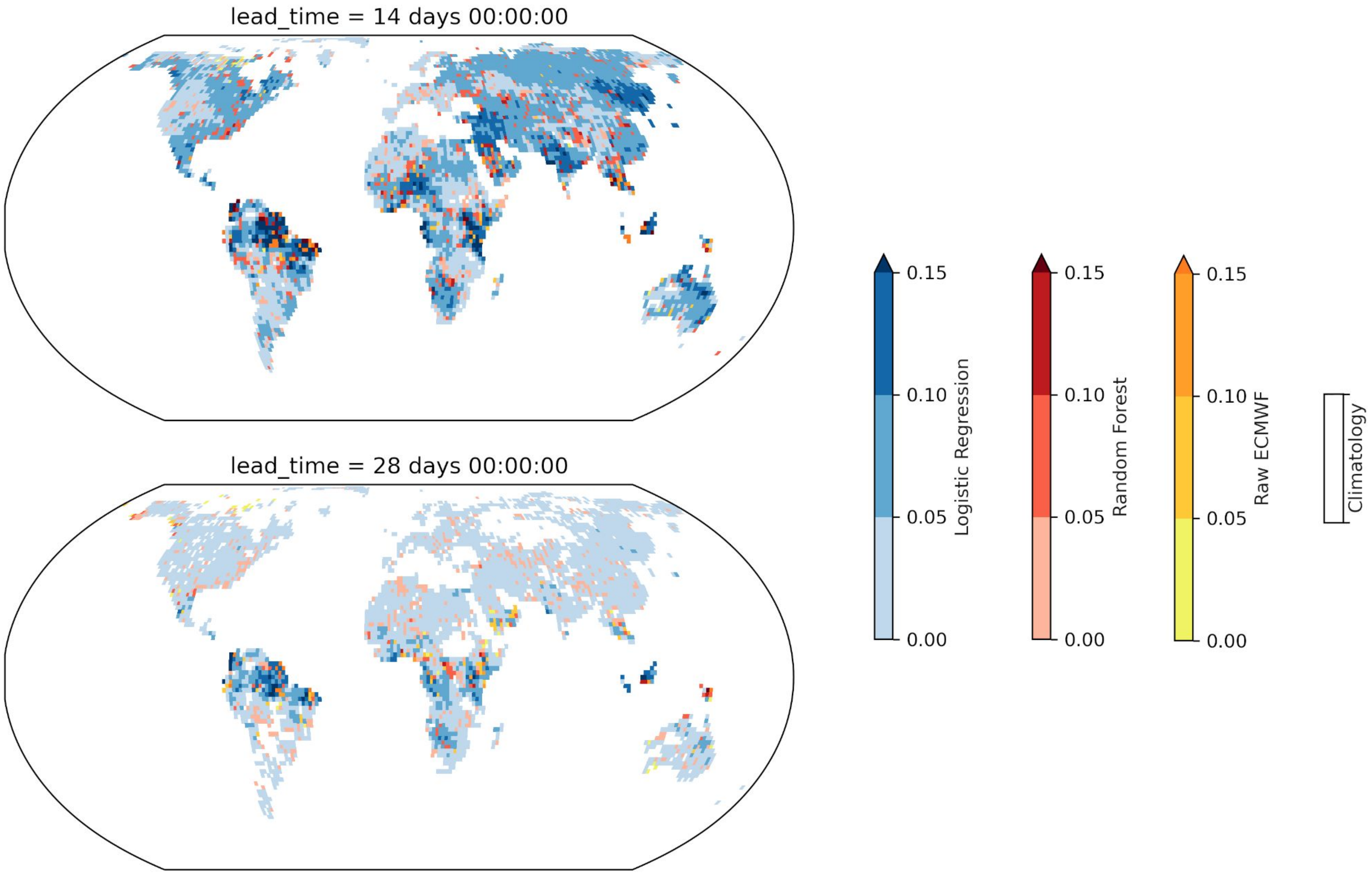
Example of a decision tree with sorted members

Random Forest: RPSS 2000-2019



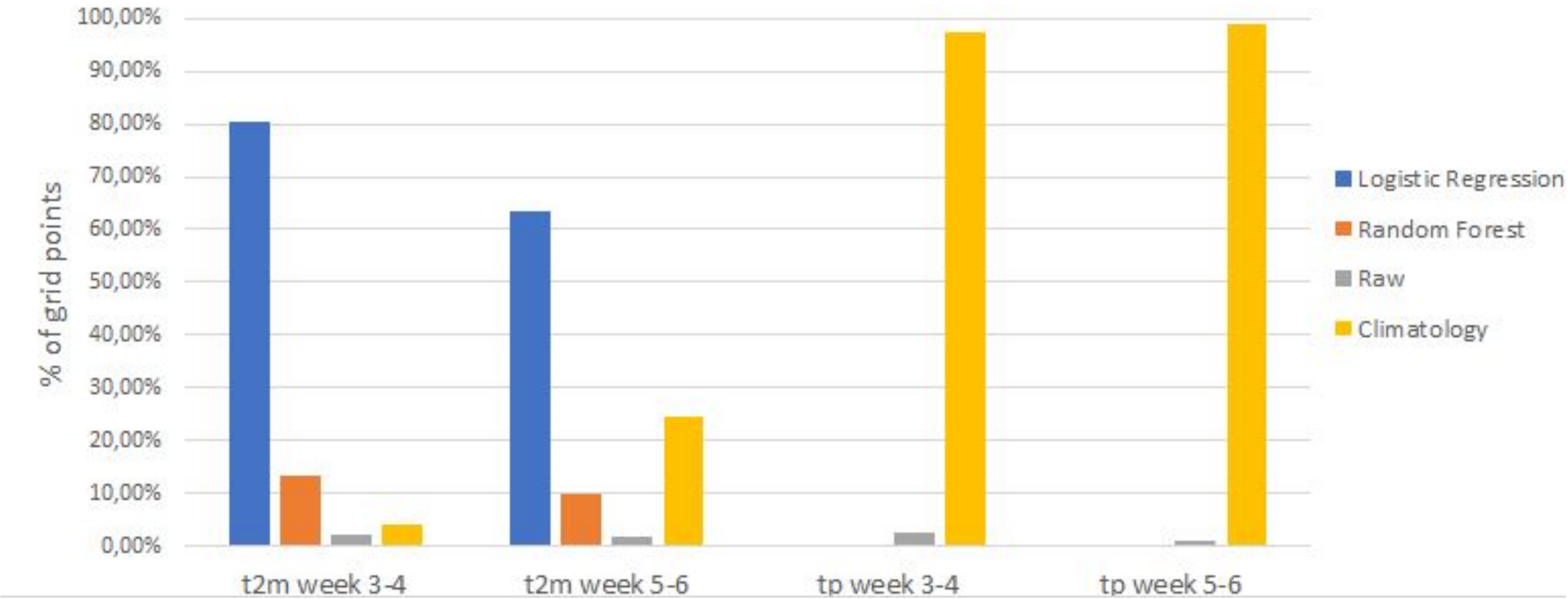
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Best model: RPSS 2000-2019



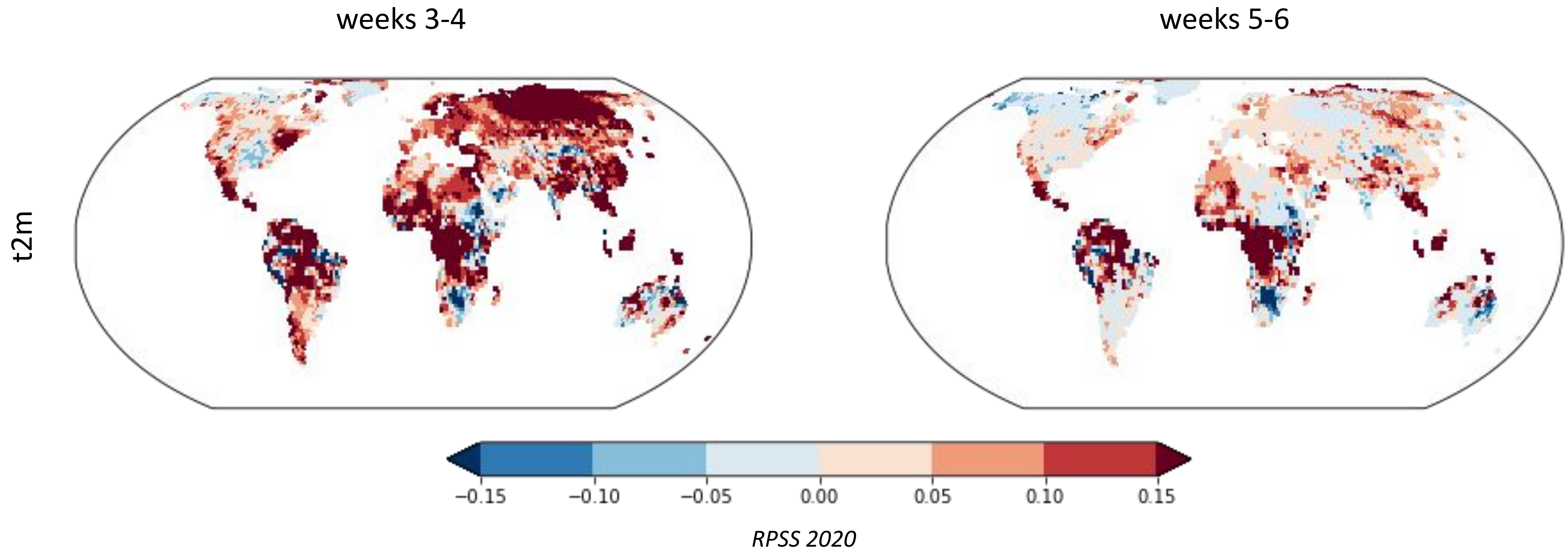
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Best model 2000-2019



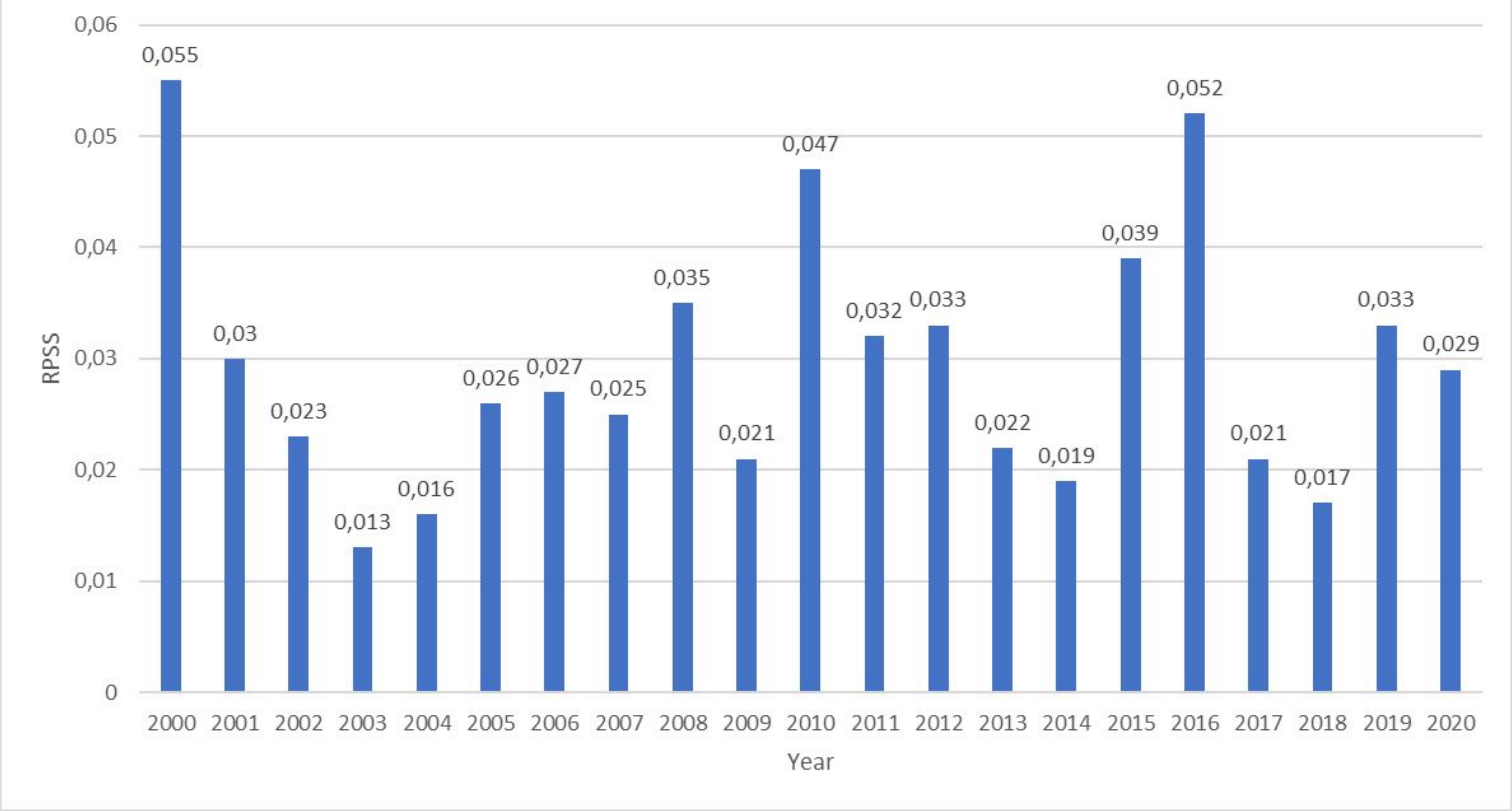
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

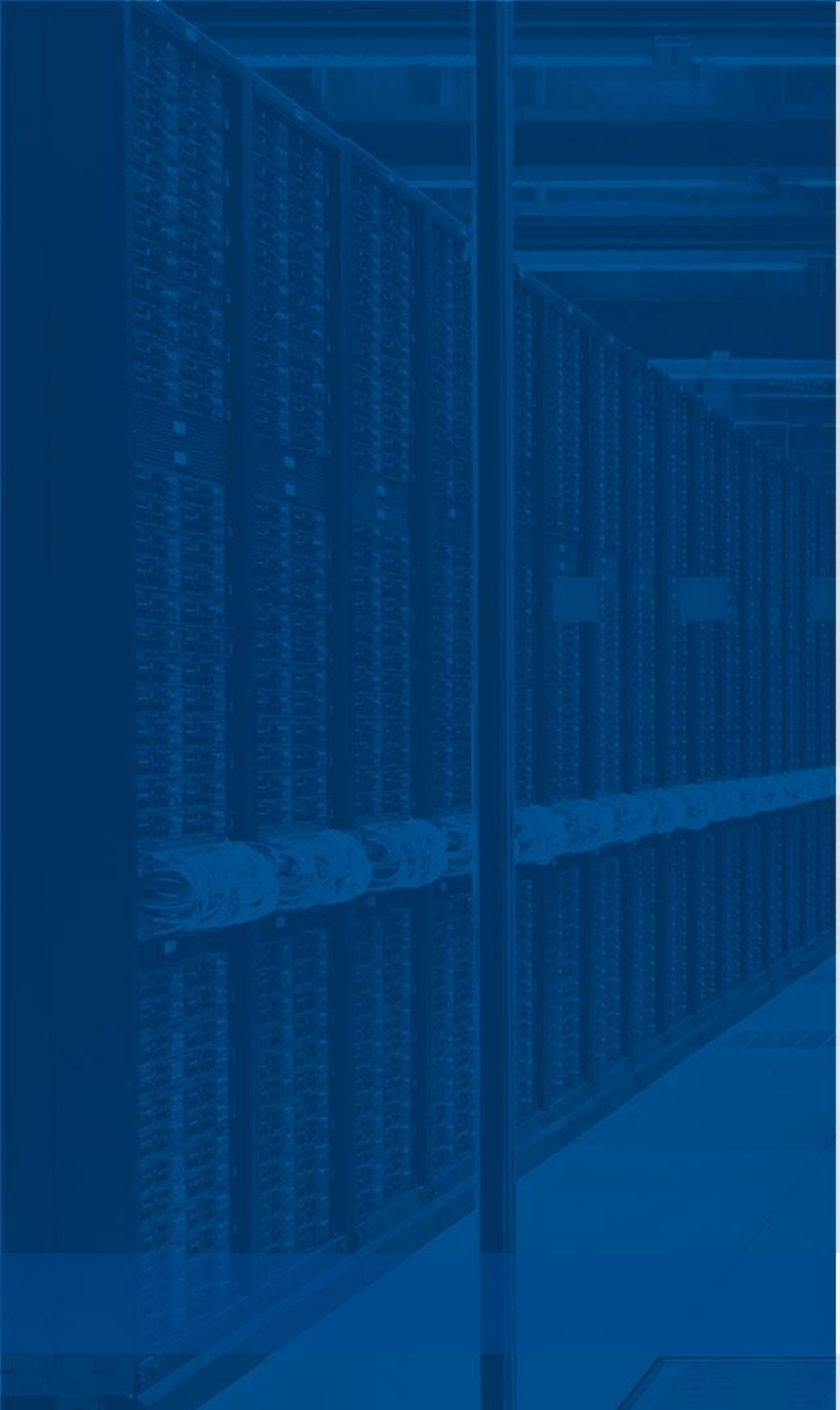
Best method: RPSS 2020



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

RPSS by year (best method)





Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

More info here:

https://renkulab.io/gitlab/lluis.palma/s2s-ai-challenge-bsc/-/blob/master/notebooks/BSC_contribution.ipynb

llledo@bsc.es