Flood map prediction challenge: phase 2 evaluation

Cyril Lemaire

March 14, 2024

1 Evaluation process

During this phase, the selected finalists are asked to improve their models on aspects other than performance. The complete codes (notebooks and or scripts) will be submitted to a committee for evaluation. The codes should take the provided datacube as an input and output a prediction vector (like during phase 1). The different steps of the pipeline (data processing, model training, model analyses, and inference) will evaluated, and the finalists will be ranked according to the criteria of trustworthy AI (detailed in the following sections). Some criteria will be measured quantitatively (performance, calibration, frugality) the rest will be evaluated on the quality of your approach. For that reason, **the winner of the challenge might not be the participant with the highest AUC ROC.**

The codes should be open-sourced on a dedicated public GitHub page. The submission should be sent by mail at this address: hackathonfloodmapping.fr@capgemini.com, before the end of the last day of the phase: April 2nd GMT+1. It should contain:

- A link to your GitHub repository containing the code.
- A document explaining how you addressed the evaluation criteria listed below. For each criterion, you should answer the listed questions to explain your approach and clearly identify the relevant portion(s) of the code. You can add any additional information.

After that, the organizing team will run all the codes on the same machine (24 CPUs, 128 Go RAM, GPU with 48GO vRAM) and evaluate the submissions during the following 2 weeks. The results will then be announced and the 3 prizes awarded.

Don't hesitate to ask questions by mail or on the forum.

2 Performance Evaluation

2.1 Model performance: AUC ROC

The performance of the model will be evaluated on a significant sample of the secret set using the **AUC ROC**. As a different sample of the set was used for phase one, the performance of your model for phase 2 might slightly differ from the one on the leaderboard.

- *Explain how you optimize the performance of your model. What methods did you use to process your data, select your features, choose your model and hyperparameters?*

2.2 Use case performance: False positive rate

To check that the model doesn't predict too many floods that are not happening, will measure the **false positive rate**. In this contexte we define a false positive by a prediction score $y_{pred} \geq 0.9$ with $y = 0$.

3 Robustness

The robustness of your model will be examined on 2 dimensions: **Calibration** and **validity domain**.

3.1 Calibration

The Calibration will be measured using the Brier Skill Score (BSS):

$$BSS = 1 - \frac{BS}{BSr},$$

in which BS is the Brier Score:

$$BS = \frac{1}{N}\sum_{t=1}^{N}(f_t - o_t)^2$$

The Reference Brier Score (BSr) is the expectation of a positive label in your entire dataset (the proportion of positive labels). f_t is the prediction score of your model, o_t is the actual outcome of the event at instance t (0 if it does not happen and 1 if it does happen) and N is the number of prediction instances.

- *Explain how you calibrated your model.*

3.2 Validity Domain

Studying the validity domain of your model allows you to have a better idea of where your model is strong and where it is not. There is no exact definition for the validity domain, however, you can get a good estimation by studying the distribution (coverage) of your training data and the distribution of performance of your model. For this use case, you can explore the validity of your model in space and time. For instance, the model can perform better for certain rivers than others, or for certain monthes.

- *Analyse the validity of your model on the provided data and give recommendations on how the model should be used (strength and limitations) for the prediction of floods.*

4 Frugality

Some machine learning models and hyperparameters allow only a marginal performance increment at the cost of a significant computational overhead. Frugality aims to reduce the overall CO2 footprint of the algorithm and cut computation times. The frugality of your solution will be measured by the time that your submission takes to run (data processing, training, and inference) on our machine (all codes will be run with the same setup).

- *Explain how you minimized the computational footprint of your solution*

5 Explainability

Model Explainability allows the data scientist to better understand the behavior of the model and the user to make better use of its predictions. You can use any ExplanableAI libraries available to explain the predictions of your model.

- *What features are important for your model and how do they influence predictions?*
- *Using explainability: evaluate the coherence of your model with your understanding of flooding mechanisms. Can you provide insights on flooding patterns and causes that could be useful for the user?*