

Hands-on tutorial on how to create your own challenge or benchmark

Adrien Pavão

LISN, CNRS

Université Paris-Saclay

France

ADRIEN.PAVAO@GMAIL.COM

Reviewed on OpenReview: *No*

Abstract

Organizing a challenge allows you to crowd-source the most difficult machine learning problems. It is also an excellent way to learn data science. By following this short hands-on tutorial, you can create your first competition or benchmark — as early as today! In this chapter, we give you everything you need to implement, concretely, your own online competition or benchmark. We do not address other practical issues such as finding sponsors or communicating about the event; this is discussed in chapter 13.

Keywords: tutorial, CodaLab, Codabench

1 Introduction

In this chapter, you will learn how to organize, a challenge or a benchmark on the two platforms *CodaLab Competitions* and *Codabench* (Figure 1). This tutorial is divided into three parts: we first review the aspects shared by both platforms (Section 2), then the *CodaLab Competitions* platform specificity (Section 3), and finally the *Codabench* platform specificity (Section 4).

CodaLab Competitions Pavao et al. (2022) is an open-source web platform hosting data science and machine learning competitions. This means that you can set up your own instance of it, or use the main instance on codalab.lisn.fr. *CodaLab* puts an emphasis on science and each year hundreds of challenges are organized on it, pushing the limits in many areas: physics, medicine, computer vision, natural language processing or even machine learning itself. Its flexibility allows hosting challenges on a wide variety of tasks! The only limit is your imagination.

Codabench Xu et al. (2022) is another project, free and open-source as well, following the steps of *CodaLab Competitions*. It can be seen as an upgrade of it, using more recent technologies, and with an emphasis on benchmarks. This emphasis on benchmark is enforced by some features, such as the possibility of filling a leaderboard with a single user account. The public server is codabench.org.

These two platforms are well suited for a tutorial, given their flexibility and the fact that they are open-source and free to use. **Once you have an account, you can already publish your first competition or benchmark!**



Figure 1: Sources: codalab.lisn.fr, codabench.org

2 General aspects

Inside the competition bundle

To create a machine learning challenge or benchmark on these platforms, all you need to do is to upload a **competition bundle**. A competition bundle is a ZIP file containing all the pieces of your competition: the data, the documentation, the scoring program and the configuration settings. To customize your competition, you can simply change the files contained in the template bundle before uploading it. Note that every aspects of the competition (settings, data, etc.) can still be edited after the upload. Let's have a closer look at what's inside the bundle.

The competition.yaml file defines the **settings** of your challenge. The title, description, logo, dates, prizes, Docker image¹, leaderboard structure and so on. All possible settings are documented in the Wiki.

The documentation files, either *HTML* or *Markdown* files, define the various pages that participants can see when going to your competition. Use them to provide the documentation and the rules, as well as any information you find important. You can of course select your own **logo** for the competition by replacing the "logo.png" file.

Data. If you are designing a machine learning problem, it is likely that you have data. The **public data** folder is for the data fully accessible by participants, **input data** is accessible by participants' submitted code, and the **reference data** folder is for storing the ground truth information (e.g. the labels from the testing set) which is kept hidden to the participants, making it only accessible by the scoring program, as explained later in this section. You can either use a data format provided in a competition example or a new one that fits well with your competition. To ensure compatibility, you will need to update the scoring program — we will talk about it in the next section. *If your problem does not involve data, don't worry! CodaLab is flexible and allows you to define any kind of problem (e.g. reinforcement learning tasks).*

The ingestion and scoring programs are the critical pieces of your competition bundle since they define the way submissions will be executed and evaluated respectively. If you want to allow only **result submissions**, then you only need the scoring program; the ingestion program is useful for **code submissions**. Figure 2 shows the interactions between the submissions (results submissions or code submissions), the programs and the leaderboard.

- The **ingestion program** defines how to train the models and save their predictions.
- The **scoring program** defines how to compare the predictions with the ground truth and computes a score.

¹The Docker image is the environment in which all submissions will be run, allowing to precisely control the evaluation procedure. It can be referred by its DockerHub name.

These programs evaluating submissions can be customized by the organizers to adjust them to adapt to any competition protocol. While they are written in Python in the templates provided, they can be written in any programming language.

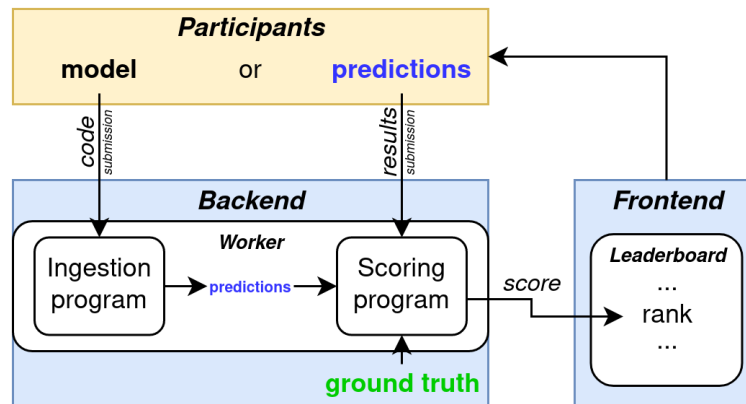


Figure 2: General competition workflow on CodaLab

The starting kit. If you have already joined a challenge as a competitor, you probably know how important it is to have a good starting kit. The goal of the starting kit is to provide participants with all the necessary resources to facilitate their dive into your competition, such as some example submissions, Jupyter notebooks, or any useful documentation and files. You can even provide the competition bundle itself (without the ground truth) inside the starting kit; this way, all the internal functioning will be perfectly transparent for the participants.

Queues and Docker

The public servers provide default compute workers. However, to run computationally demanding competitions, **organizers can create custom queues and attach their own CPU or GPU compute workers** (physical or virtual machines on any cloud service) to it. This modular architecture of *CodaLab Competitions* has been a key ingredient in growing its user base, without requiring that the institution hosting the main instance covers all computational costs. Another interesting aspect of this feature is that the training and testing of algorithms can be done on confidential data, without any leakage, by putting data directly inside the compute workers. This is especially useful for medical research, challenges organized by industries, and in other restricted domains.

The workflow of the jobs, showed in Figure 3, works as follows: each competition can be linked to only one queue, but the queue can be used by several competitions. Each worker can listen to only one queue, but the queue can be linked to several workers.

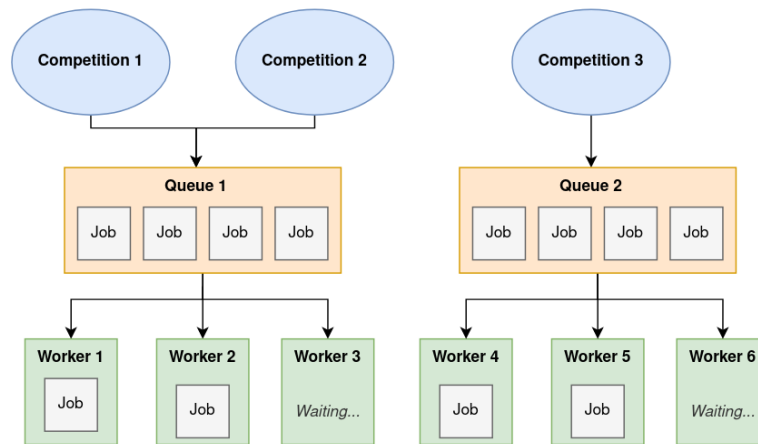


Figure 3: Diagram of the structure of workers and queues. The queues dispatch the jobs between the compute workers. Note that a queue can receive jobs (submissions) from several competitions, and can send them to several compute workers.

To setup a machine as a compute worker, you need to install Docker, note down the URL of your queue, and run a single command line²³.

The number of workers can be adjusted at anytime. This means that you can add more workers to the queue during the competition, they will dispatch all jobs automatically, increasing your computing power in real time in order to fit the needs of your competition.

One aspect that allows the same machine to compute jobs from many different competitions is the use of **Docker environments**. The execution of participants code and scoring are performed inside a container, which prevents to damage the servers and allows organizers to define a custom controlled environment for their competitions. Organizers can create a fully customized environment, with allowed libraries and programming languages for their participants' submissions, and simply link it to a competition by providing a DockerHub name and tag. This means that every candidate is judged in the same way, the competition does not get deprecated after some time and adding new libraries or updating the experimental environment is straightforward and transparent.

Organizer features

As a competition or benchmark creator, you have access to useful organizer-only features. These features are accessible from the grey buttons at the top of the user interface, as shown in Figure 4.

²https://github.com/codalab/codalab-competitions/wiki/User_Using-your-own-compute-workers

³<https://github.com/codalab/codabench/wiki/Compute-Worker-Management---Setup>

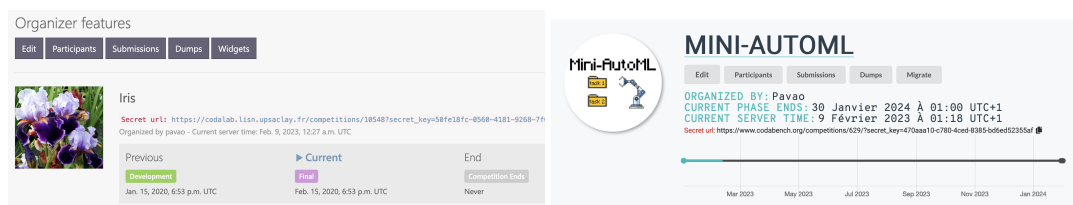


Figure 4: As an organizer, you have access to many interesting features, accessible from the grey buttons at the top of the interface of *CodaLab Competitions* (left) or *Codabench* (right).

Editor. The editor allows organizers to edit a competition that is already up and running. From this panel, you can edit every setting, and replace the data, scoring program or ingestion program.

Manage participants. You can choose to allow anybody to join your competition, or to have a registration process and validate who can join. You can accept or revoke access of the participants at any time.

Manage submissions. A panel to manage all the submissions made by the participants. From this panel, you can access details about the submissions: their date and author, the output and logs of the scoring programs and the submissions themselves (the files uploaded by the participants). The interface can be used to cancel, remove or re-run submissions. Overall, it is very useful to debug, to prevent cheating or to run post-challenge analysis.

Dumps: export your competition. A feature that you can use to download your competition as a bundle. All changes made directly through the editor will be saved and the resulting bundle can be re-uploaded on the platform or on any other instance of it. If you wish to re-upload the bundle on the same platform and want to keep the bundle as light as possible, you can use the option “use URI keys instead of files”, in this case the datasets and programs will be referred by their address in the storage.

Publish your competition. By default, your competition is private. When a competition is private, it can be accessed only by its administrators, or by anyone from the “secret URL”. Once you publish your competition, anyone can access it from the public URL. You can make your competition private again at any time.

Auto-migration. Automatically re-run the leaderboard’s submissions from one phase to another phase.

3 CodaLab Competitions tutorial

Get started

To create your first machine learning challenge, all you need to do is to upload a **competition bundle**. A competition bundle is a ZIP file containing all the pieces of your competition: the data, the documentation, the scoring program and the configuration settings, as explained in Section 2. Let’s start from an example; it’s the easiest way. Here is the competition bundle of the *Iris Challenge*, based on the famous Fisher’s dataset Fisher (1936): *Iris Competition Bundle*. Now, go to CodaLab, and upload the file named “iris_competition_bundle.zip” as shown in Figure 5. Go to “My Competitions”, then “Competitions I’m Running”, and finally “Create Competition”. During the last step you will be redirected to the final menu where you can upload the competition

bundle. After uploading it, you will see the message of Figure 6 indicating that your competition is successfully created and ready to receive submissions.

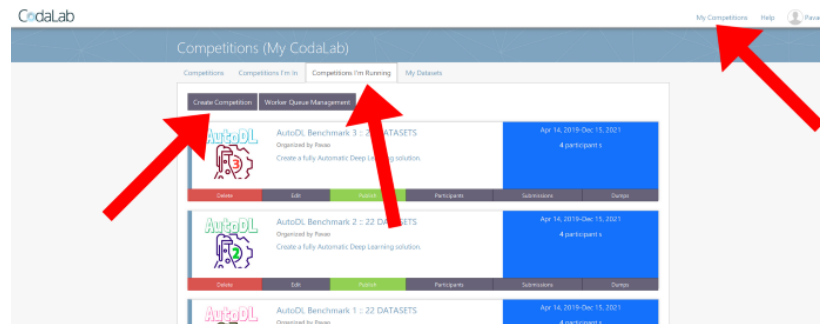


Figure 5: Go to “My Competitions”, then “Competitions I’m Running” and finally “Create Competition”.

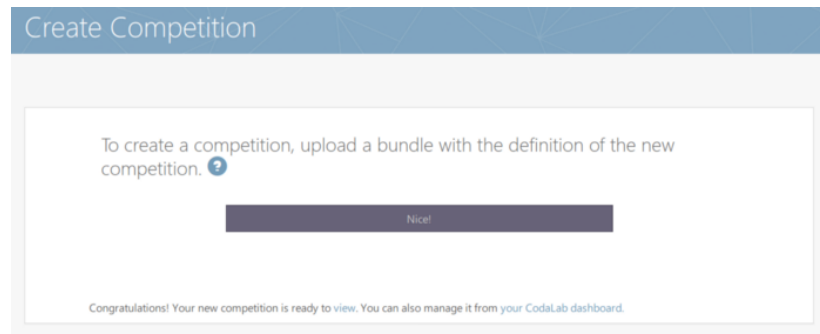


Figure 6: That’s it! Your competition is ready to receive submissions.

Once the competition is uploaded, you can begin to make submissions. To do so, UnZip the downloaded bundle, go inside the starting kit and zip the content of either the “sample_result_submission” or the “sample_code_submission” folder. It is important to zip the files without directory structure and to include the “metadata” file in the case of code submission. Then, on the website (see Figure 7), go to the “Participate” tab, then “Submit / View results”, click on “Submit” and select your zip file. The submission will process for a few moments. After that, you’ll be able to see your score in the leaderboard, in the “Results” tab.

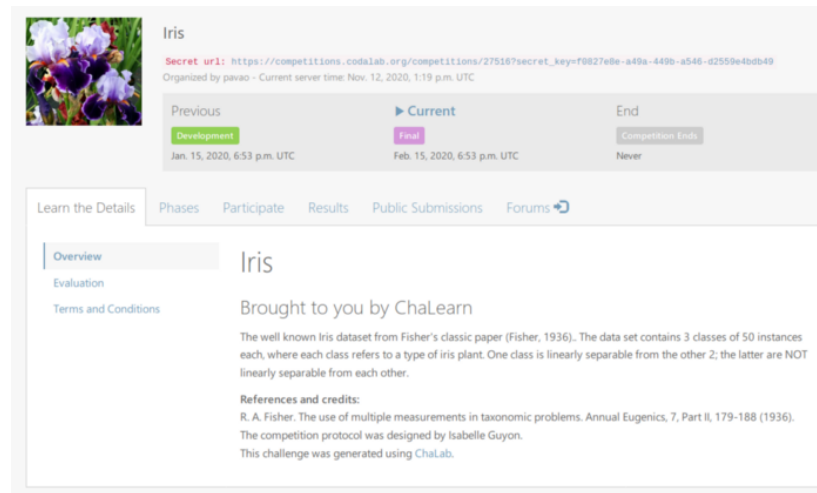


Figure 7: Main page of the Iris Challenge. The web pages are defined by the HTML files from the bundle.

Customize competition

Your competition is up and running! However, you wish to edit it. It is still possible. As an administrator of your own challenge, you have access to the “Edit” menu: a panel in which you can edit every setting. Here are some examples:

- Force submission to leaderboard: if enabled, the last submission of a participant is the one showed on the leaderboard.
- Disallow leaderboard modifying: if disabled, users can select which of their submissions appear on the leaderboard.
- Share administrator rights: you can add *CodaLab* users as administrators of your competition by giving their username or email address. They’ll have access to all organizer-only features, except deleting the competition.
- Anonymous leaderboard: if enabled, the username are hidden in the leaderboard.
- Registration required: if enabled, users need to request the access to your competition. You then have to accept or reject their participation manually from the “Participants” tab.
- Select the queue.
- Specify competition docker image by its DockerHub name.
- For each phase you can chose different data and scoring programs.

If you wish to change the dataset or the scoring program, you’ll first need to upload the new version from the “*My Datasets*” page, as shown in Figure 8. You will then be able to select it in the editor.

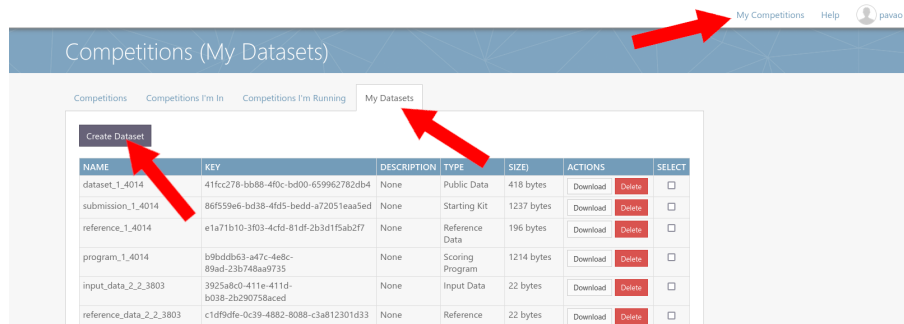


Figure 8: Go to “My Datasets” to upload new datasets or programs.

4 Codabench tutorial

Get started

As an evolution of *CodaLab Competitions*, *Codabench* is similar in terms of functioning and features. In this part of the tutorial, we will organize a simple benchmark on *Codabench*, in which the participants can have multiple entries displayed on the leaderboard. Note that competition bundles from *CodaLab Competitions* are compatible with *Codabench*, while not vice versa.

Let’s have a look at the Mini-AutoML Bundle. To upload the file named “bundle.zip” to *Codabench*, go to “Benchmarks”, then “Management” and finally “Upload” (Figure 9). Then click on the paper clip button to select and upload the bundle (Figure 10).

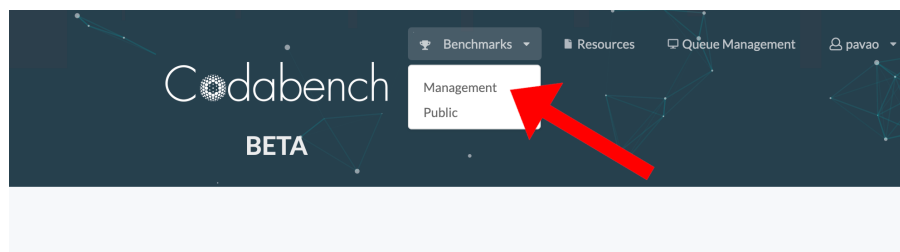


Figure 9: Go to “Benchmarks”, then “Management” and finally “Upload”.



Figure 10: Then click on the paper clip button to select and upload the bundle.

The main page of the benchmark should look like Figure 11.

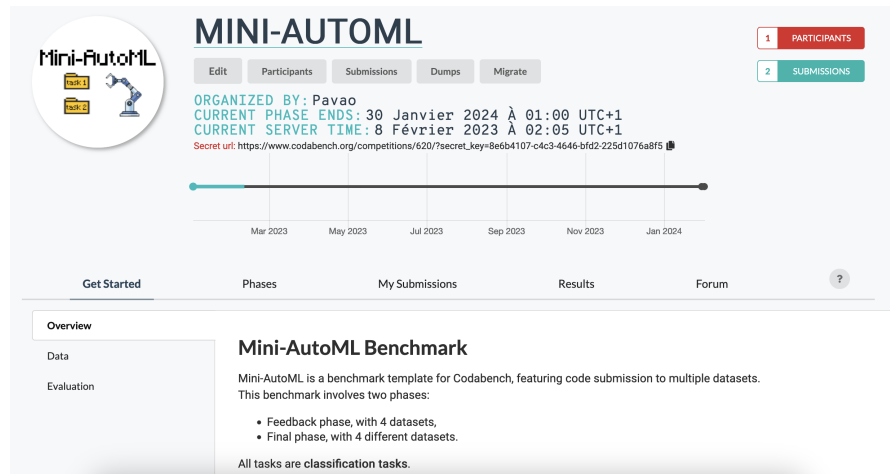


Figure 11: Overview of Mini-AutoML main page.

Let's now make a submission. To do so, download the file `sample_code_submission.zip` and upload it in the “*My Submissions*” tab of the benchmark. You will be able to view logs in real time during the processing of the submission. The leaderboard, available in the “*Results*” tab of the benchmark, will then be updated.

Customize benchmark

Let's customize this newly created benchmark, with the goal in mind to be able to fill in the leaderboard with many different models, in order to compare them.

Fact sheets. In the first tab of the editor, you can enable “fact sheets” to gather more information about the submissions. Enabling fact sheets means that the participants will be asked to fill in some information when making submissions. You can fully customize the information fields, making them required or not, and choosing which information appears on the leaderboard. This can be used to display the name of the methods, the URL to the source code, or a description of the method. The gathering of metadata about the methods used is crucial when conducting a benchmark, and this interface makes it simple to gather all this information in one place.

Edit data and programs. *Codabench* provides an interface to upload *Ingestion Program*, *Scoring Program*, *Starting Kit* and *Data* (*public data*, *input data* and *reference data*). To upload a new dataset or program, go to “*Resources*”, “*Datasets*” and click on “*Add Dataset*”. Then name it, select your ZIP file, and chose the type (reference data, scoring program, etc.) (Figure 12).

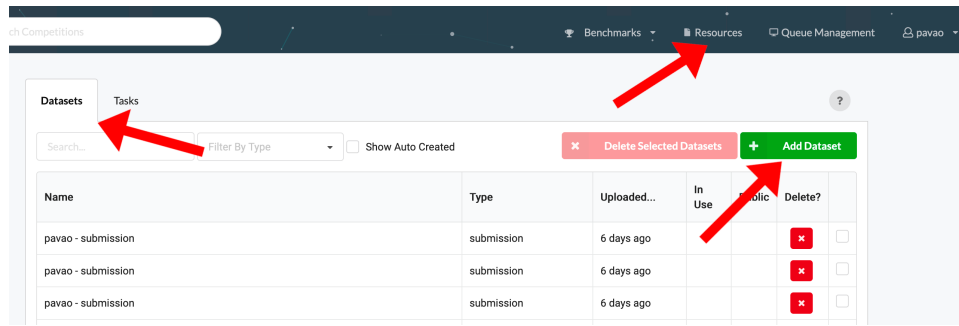


Figure 12: To upload a new dataset or program, go to “Resources”, “Datasets” and click on “Add Dataset”. Then name it, select your ZIP file, and chose the type (reference data, scoring program, etc.)

Once these are uploaded, a task can be created using the uploaded files. A task is a combination of *Ingestion Program*, *Scoring Program*, *Input Data*, and *Reference Data*. To create a task, go to “Resources”, “Tasks” and click on “Create Task”. Then name it and select the files previously uploaded: Input data, reference data, ingestion program and/or scoring program (Figure 13).

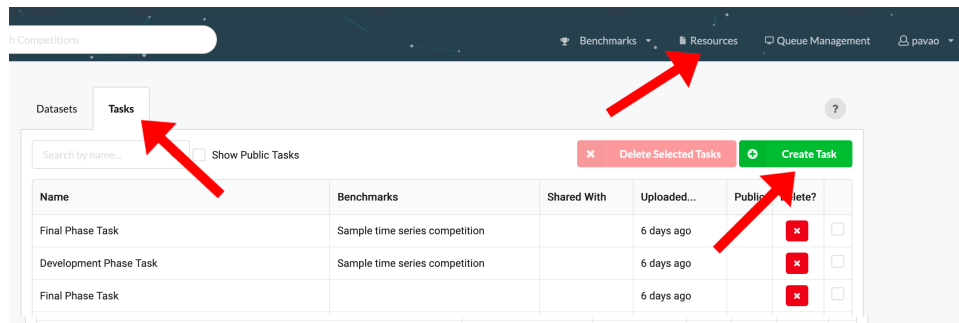


Figure 13: To create a task, go to “Resources”, “Tasks” and click on “Create Task”. Then name it and select the files previously uploaded: Input data, reference data, ingestion program and/or scoring program.

In your competition editor, you can add, update or delete a task for each phase. Unlike *CodaLab*, *Codabench* allows you to have multiple tasks for each phase. To associate a task to a phase, go to the editor, then “Phases”, click on the edit button and select the desired task (Figures 14 and 15).

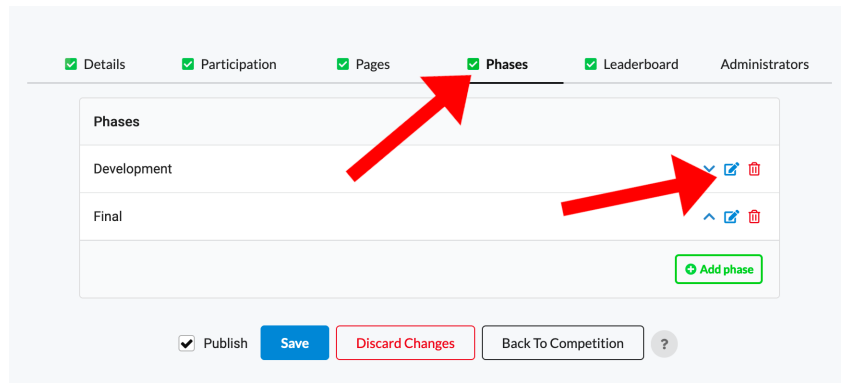


Figure 14: Go to the editor, then “*Phases*” and click on the edit button.

Figure 15: You can now associate the desired task to the phase.

Submission rules. The submission rule programs the behavior of the leaderboard regarding new submissions. Submissions can be forced to the leaderboard or manually selected, can be unique or multiple on the leaderboard, etc. To edit the submission rule, go to the editor, then “Leaderboard” and edit the leaderboard (Figure 16). Then change the submission rule from “Force Last” to “Add And Delete Multiple” (Figure 17). “Force Last” means that only the last submission of each participant will be shown on the leaderboard, while “Add And Delete Multiple” means that the participants will be allowed to manually select multiple submissions to show on the leaderboard.

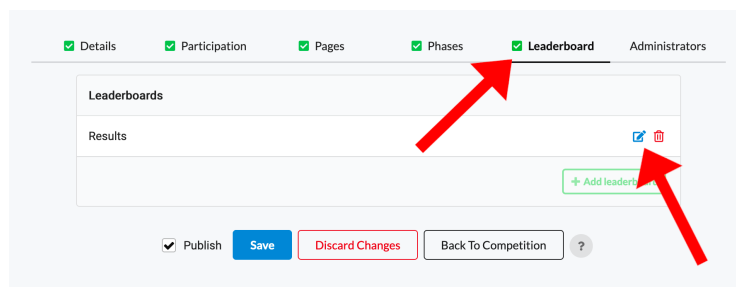


Figure 16: To edit the submission rule, go to the editor, then “Leaderboard” and edit the leaderboard.

The screenshot shows the 'Leaderboard form' interface. Under 'Leaderboard Settings', the 'Title' is 'Results' and the 'Key' is 'Results'. The 'Submission Rule' dropdown is set to 'Add And Delete Multiple', highlighted by a red arrow. Below this, there are columns for 'Average Accuracy', 'Dataset 1', 'Dataset 2', 'Dataset 3', and 'Dataset 4'. The 'Primary Column' is set to 'Average Accuracy'. The 'Computation' dropdown is set to 'Average'. The 'Apply to' dropdown is set to 'Dataset 1'. At the bottom, there are buttons for '+ Add column', 'Cancel', and 'Save'.

Figure 17: Change the submission rule from “Force Last” to “Add And Delete Multiple”.

The current submission rule, “Force Last”, means that only the last submission of each participant will appear on the leaderboard. This is a classical setting for competition. Changing this rule to “Add And Delete Multiple” will allow the participants to manually select which submissions will appear on the leaderboard, and multiple submissions per participant on the leaderboard are allowed.

Add submissions to the leaderboard. Now that the leaderboard is set up, let’s submit different variations of the code of the model from the `sample_code_submission.zip`. This example code submission simply calls a classifier from Scikit-Learn Pedregosa et al. (2011). Replace the *DecisionTreeClassifier* with the classifier of your choice. Remember to differentiate the different submissions by filling the “Method name” in the fact sheet. Once your submission is processed, click on the leaderboard button under “Actions” in the submissions table to manually add them to the leaderboard (Figure 18).













ID #	File name	Date	Status	Actions
4401	knn.zip	2023-02-09 02:16	Finished	 
4400	mlp.zip	2023-02-09 02:16	Finished	 
4399	gaussiannb.zip	2023-02-09 02:16	Finished	 
4398	rf.zip	2023-02-09 02:14	Finished	 
4397	rf.zip	2023-02-09 02:14	Finished	 
4395	sample_code_submission.zip	2023-02-08 16:30	Finished	 

Figure 18: Once your submission is processed, click on the leaderboard button under “Actions” in the submissions table to manually add them to the leaderboard.

The leaderboard finally looks like Figure 19.

Results								
Task:		Fact Sheet Answers	Development Task					
#	Participant	Method name	Average Accuracy	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Duration
1	Pavao	Random Forest	0.8974364460	0.8551832389	0.7678958785	0.9666666667	1.0000000000	2.9789698124
2	Pavao	K-Nearest Neighbors	0.8711441508	0.7730840101	0.7592190889	0.9555555556	0.9967179487	1.4459712505
3	Pavao	MultiLayer Perceptron	0.8709400671	0.7963898178	0.6984815618	0.9888888889	1.0000000000	3.9009172916
4	Pavao	Decision Tree Baseline	0.8622140842	0.8106189859	0.6832971800	0.9555555556	0.9993846154	0.2303986549
5	Pavao	Gaussian NB	0.8496961070	0.7948201733	0.7678958785	0.9222222222	0.9138461538	0.1322979927

Figure 19: Screenshot of the filled up leaderboard. The random forest classifier did the best job in the first phase of the benchmark!

5 Conclusion

Congratulations! You have learned the basics of *CodaLab Competitions* and *Codabench*, and now you can organize your own competitions or benchmarks! However, we barely scratched the surface of all the possibilities offered by these platforms. To learn more, you can refer to: CodaLab Competitions Documentation and *Codabench* Documentation.

From the documentation, you will learn how to link your personal compute workers (CPU, GPU), how to customize the ingestion and scoring programs, how to define complex leaderboards with multiple criteria, or even how to deploy your own instance of the platform. You can also join the effort and develop your own features!

References

- R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annual Eugenics*, 7(Part II):179–188, 1936. The competition protocol was designed by Isabelle Guyon. This challenge was generated using ChaLab for CodaLab v1.5.
- Adrien Pavao, Isabelle Guyon, Anne-Catherine Letournel, Xavier Baró, Hugo Escalante, Sergio Escalera, Tyler Thomas, and Zhen Xu. CodaLab competitions: An open source platform to organize scientific challenges. *Technical report*, 2022. URL <https://hal.inria.fr/hal-03629462v1>.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- Zhen Xu, Sergio Escalera, Adrien Pavao, Magali Richard, Wei-Wei Tu, Quanming Yao, Huan Zhao, and Isabelle Guyon. Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform. *Patterns*, 3(7):100543, 2022. ISSN 2666-3899. doi: <https://doi.org/10.1016/j.patter>.

2022.100543. URL <https://www.sciencedirect.com/science/article/pii/S2666389922001465>.