Journal of Machine Learning Research 24 (2023) 1-6 Submitted 12/21; Revised 2/23; Published 6/23

CodaLab Competitions
An Open Source Platform to Organize Scientific Challenges

Adrien Pavão[1,2] ADRIEN.PAVAO@GMAIL.COM
Isabelle Guyon[1,2] GUYON@CHALEARN.ORG
Anne-Catherine Letournel[1,2] ANNE-CATHERINE.LETOURNEL@LISN.UPSACLAY.FR
Dinh-Tuan Tran[1] DINH-TUAN.TRAN@LISN.UPSACLAY.FR
Xavier Baró[3] XBARO@UOC.EDU
Hugo Jair Escalante[2,4] HUGOJAIR@INAOEP.MX
Sergio Escalera[2,5] SERGIO@MAIA.UB.ES
Tyler Thomas[6] MR.TYLER.THOMAS@GMAIL.COM
Zhen Xu[7] XUZHEN@4PARADIGM.COM
[1] *LISN, CNRS, Université Paris-Saclay, France,* [2] *ChaLearn,*
[3] *Universitat Oberta de Catalunya, Spain,*
[4] *Instituto Nacional de Astrofísica, Óptica y Electrónica, Mexico,*
[5] *Universitat de Barcelona and Computer Vision Center, Spain,*
[6] *Tier0 Software Development LLC, United States of America,*
[7] *4Paradigm, China*

Editor: Antti Honkela

Abstract

CodaLab Competitions is an open source web platform designed to help data scientists and research teams to crowd-source the resolution of machine learning problems through the organization of competitions, also called challenges or contests. *CodaLab Competitions* provides useful features such as multiple phases, results and code submissions, multi-score leaderboards, and jobs running inside Docker containers. The platform is very flexible and can handle large scale experiments, by allowing organizers to upload large datasets and provide their own CPU or GPU compute workers.

Keywords: competition, challenge, contest, platform, code submission, crowdsourcing

1. Introduction

Data science competitions have become an established mechanism for solving theoretical and practical problems in machine learning and related fields. Throughout history, major achievements have arisen from challenges, ranging from the 1714 Longitude Prize (Quill and Penney, 1976) to the ImageNet challenge (Russakovsky et al., 2015); the latter witnessing the most recent renaissance of deep learning in computer vision. Given the relevance and potential impact of competitions, the availability of off-the-shelf platforms allowing anyone to create and run their own competition is a *catalyzer* for making rapid progress while ensuring inclusiveness and building community.

This contribution features *CodaLab Competitions*[1], an open source platform for running data science competitions that has been used in hundreds of challenges associated to physics, machine

[1] The platform is available at `https://codalab.lisn.fr/`.

©2023 Adrien Pavão, Isabelle Guyon, Anne-Catherine Letournel, Dinh-Tuan Tran, Xavier Baró, Hugo Jair Escalante, Sergio Escalera, Tyler Thomas, Zhen Xu.

License: CC-BY 4.0, see `https://creativecommons.org/licenses/by/4.0/`. Attribution requirements are provided at `http://jmlr.org/papers/v24/21-1436.html`.

Criteria	AICrowd	**CodaLab Competitions**	CrowdAnalytiX	EvalAI	Kaggle	RAMP	Tianchi
Code-sharing	✗	✓	✗	✓	✓	✓	✓
Code submission	✓	✓	✗	✓	✓	✓	✓
Active community	☆☆	☆☆☆	☆	☆☆	☆☆☆	☆	☆☆☆
Custom metrics	✓	✓	✓	✓	✓	✓	?
Staged challenge	✓	✓	✗	✓	✗	✓	✗
Private evaluation	✗	✓	✗	✓	✗	✗	✗
Open-source	✓	✓	✗	✓	✗	✓	✗
Human evaluation	✗	✗	✗	✓	✗	✗	✗
RL-friendly	✓	✓	✗	✓	✗	✗	✗
Run for free	✗	✓	✗	✓	✓	✗	?

Table 1: Comparison of competition platforms, borrowed from Rousseau and Ustyuzhanin (2020). References for EvalAI: Yadav et al. (2019), AICrowd: Mohanty et al. (2016), CrowdAnalytiX Mishra (2012), Kaggle: Goldbloom and Hamner (2010), RAMP: Kégl et al. (2018) and Tianchi: Group (2014). Isdahl and Gundersen (2019) have compare platforms more extensively. Other platforms are presented in Neo (2021).

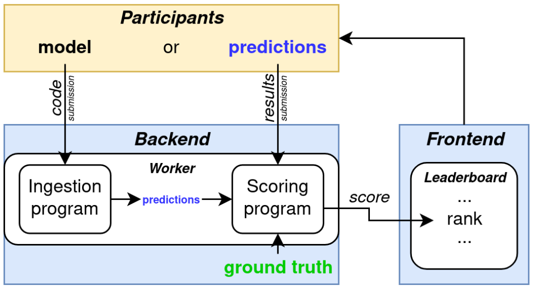

Figure 1: General competition workflow.

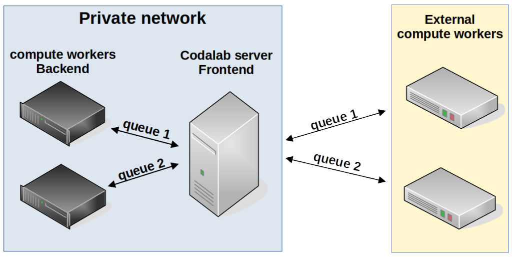

Figure 2: Compute worker connections.

learning, computer vision, natural language processing, among many other fields. *CodaLab Competitions* has unique features that make it an excellent open source option for organizing competitions. Table 1 shows a comparison of *CodaLab Competitions* with other popular platforms. The reader is referred to the project website where the code and complete documentation are found. The code is released under an Apache 2.0 License. This project has a sister project called CodaLab Worksheets, which features dynamical workflows, particularly useful for Natural Language Processing. Both projects are accessible from `https://codalab.org/` pointing to public platform instances, freely available. This paper concerns only *CodaLab Competitions*.

2. Key Concepts and Features

Competition bundle. A competition bundle is a ZIP file containing all necessary elements to create a competition: data, documentation, scoring program, and configuration settings. A new competition is created by uploading a bundle to the platform. The settings can be updated from the platform web interface via an editor. After editing, updated competition bundles can be re-downloaded from the platform for archival and sharing purposes. There is an option to download a light version of the bundle, without datasets and programs; the configuration file then directly points to database entries for such resources. This facilitates cloning competitions without having to re-upload data or code.

Results and code submission. *CodaLab Competitions* supports the submission of models' predictions to be scored; and also supports code submission, allowing to train and test the methods in a dedicated environment on the server side. Competitions with only results submissions are easy to organize and usually need few computational resources. On the other hand, code submission has exhibited many advantages, such as providing potentially powerful machines to the candidates (shared in a fair way among them) and improving reproducibility. The logic of scoring of a competition (Figure 1) is coded by the organizers in any programming language (typically Python): *CodaLab Competitions* requires at least a **scoring program** to score participants' submissions (e.g. by comparing results and ground truth), and optionally an **ingestion program** to execute a code submission in a controlled manner, with an organizer-defined API, e.g. calling a "fit" method and a "predict" method. Thus organizers have the flexibility of implementing any challenge protocol, with any data format or even data generating models, allowing them to organize reinforcement learning challenges.

Compute workers (add external computational resources). The public instance of *CodaLab Competitions* provides 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) (Figure 2). 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.

Competition Docker environment. 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 Docker Hub 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.

Phases. Competitions can be divided into several phases with different settings: different dates, data, scoring programs, etc., enabling elaborate competition design. For example, a feed-back phase with little restriction on number of submissions can be followed by a final test phase on a fresh test set (or new datasets altogether), eventually emulating the "public" and "private" leaderboards of other platforms. The last code submitted in one phase can be forwarded automatically to the next phase. Multi-phase challenges can help organizers keep participants engaged over long periods of time.

Multiple scores. The leaderboard is fully customizable and can handle several custom scoring functions. The final score used to rank participants can be computed using the average rank obtained on each sub-score.

Documentation. The documentation is organized according to stakeholders categories *participants*, *organizers*, *administrators*, and *contributors* directly on the first page of the documentation.

As a *participant*, you will find explanation on how to register, use the GUI to submit code or results. Mail communication with the organizers and a challenge forum are available. You will receive feedback on your submissions: scores, running time, plots, logs and predictions.

As an *organizer*, you are accompanied with many competition templates, from simple to elaborate, to ease the technical aspects of building a competition, and to let you concentrate on the scientific aspects. You can optionally propose a starting kit to the participants to lower the barrier of entry of beginners. You learn how to configure your competition: settings, logo, HTML pages, computer language, data organization, participant resource limitations, dates, and phases duration.

As an *administrator* of your own instance of *CodaLab Competitions*, each piece of the infrastructure is configurable and offered as a docker component. You can deploy your instance in a very flexible way concerning the sizing of your project thanks to deployment guide hints.

As an occasional *contributor*, you can discuss with the main developers via the GitHub issues and suggest pull requests to solve some of the issues you have encountered. As a regular contributor, some project management rules are used to facilitate any external contribution.

3. Technical Aspects

CodaLab Competitions is implemented in Python's Django framework, one of the most extended web application framework in Python, with constant maintenance and security updates and a huge list of plugins. The system is divided in three main blocks: Front-end, API, and workers. Front-end is implemented using the default Django template system, providing all the views users interact with. For the API, the Django Rest Framework is used. Front-end and API constitute the core of *CodaLab Competitions*, and are in charge of managing all the data entities of the system, such as users, competitions, and submissions. For data management, a PostgreSQL database is used, pointing to MinIO for file storage. Submissions are evaluated in an asynchronous manner, using computation workers. This process follows a classical producer-consumer approach, using RabbitMQ as a queue manager, and a Celery client for message management. This approach is also used for periodic tasks happening on the core system. Another key technology used is Docker. Although *CodaLab Competitions* itself can be deployed as a Docker, the main advantage of containerization is related to the computation workers. All selected technologies are open source and can be deployed in a self-hosted manner, or on common cloud providers, to organize competitions at scale. Project management is organized in an agile way for bug correction and development of new features, leveraging GitHub issues, labeling process, branch creation and merging. We welcome new contributors.

4. Conclusion and Ongoing Developments

CodaLab Competitions is a mature platform serving the machine learning community to organize competitions at scale. The open source nature of the project provides a level of scrutiny of the community necessary to give reassurance to competition organizers and participants that data and submissions are securely and fairly handled. This security is therefore mainly dependent on the actual platform deployment (network firewalls, disk encryption, etc.). *CodaLab Competitions* has a public instance free for use deployed at Université Paris-Saclay. Hosting, maintaining, and further developing the platform is funded by grants and donations. *CodaLab Competitions* has already undergone a variety of significant improvements and technology migration since inception, including diversification of storage services (cloud and private), introduction of organizer computation workers and self-managed queues, offering GPUs, and optimizing page loading. The current rapid growth of the user base will impose new mutations, particularly linked to storage management. Decentralization of storage, imitating the computation worker decentralization, is under study. We are also experimenting with other means of managing growth by federating various instances of the platform via single point of entry (see Chahub beta). The needs of users are also evolving and there is a demand of hosting benchmarks, with more emphasis on collaborative and reproducible research that individual achievements. A sister platform more geared to benchmarks is under development: Codabench Xu et al. (2022). The project roadmap also includes getting accreditation to host sentitive medical data.

Acknowledgments

The on-going development of the project is sustained by a community governance. The openness of *CodaLab Competitions* is total: the Apache 2.0 licence is used, the source code is on GitHub; the development framework and all the used components are open source. *CodaLab Competitions* has received important contributions from many people who did not co-author this paper, and we would like to thank their efforts in making *CodaLab Competitions* what it is today: Pujun Bhatnagar, Justin Carden, Eric Carmichael, Richard Caruana, Francis Cleary, Laurent Darré, Xiawei Guo, Ivan Judson, James Keith, Lori Ada Kilty, Shaunak Kishore, Stephen Koo, Percy Liang, Zhengying Liu, Zhenwu Liu, Pragnya Maduskar, Simon Mercer, Arthur Pesah, Christophe Poulain, Lukasz Romaszko, Loïc Sarrazin, Laurent Senta, Lisheng Sun, Dinh-Tuan Tran, Bailey Trefts, Nic Threfts, Sebastien Treguer, Wei-Wei Tu, Cedric Vachaudez, Evelyne Viegas, Paul Viola, Erick Watson, Tony Yang, Flavio Zhingri, DinhMichael Zyskowski. The project was originally funded by Microsoft Research and was hosted by the OuterCurve foundation. Université Paris-Saclay is now hosting the main instance of *CodaLab Competitions*. Funding and support have been received by Microsoft, Google, and ChaLearn, and several research grants, including Big Data Chair of Excellence FDS Paris-Saclay, United Health Foundation INCITE project, EU EIT projects HADACA and COMETH, Paris Région Ile-de-France, and ANR Chair of Artificial Intelligence HUMANIA ANR-19-CHIA-00222-01.

Authors' Contributions

CRediT Taxonomy is used for authors' contribution statement:

Adrien Pavao: Investigation, Resources, Validation, Writing-original draft

Isabelle Guyon: Conceptualization, Funding acquisition, Methodology, Resources, Supervision, Writing – review and editing

Anne-Catherine Letournel: Data curation, Project administration, Resources, Validation, Writing-review and editing

Dinh-Tuan Tran: Data curation, Investigation, Software

Xavier Baró: Investigation, Validation, Writing-review and editing

Hugo Jair Escalante: Investigation, Validation, Writing-review and editing

Sergio Escalera: Investigation, Validation, Writing-review and editing

Tyler Thomas: Software, Validation

Zhen Xu: Investigation, Validation, Writing-review and editing

References

Anthony Goldbloom and Ben Hamner. Kaggle. 2010. URL `https://www.kaggle.com/docs/competitions`.

Alibaba Group. Tianchi. 2014. URL `https://tianchi.aliyun.com/`.

Richard Isdahl and Odd Erik Gundersen. Out-of-the-box reproducibility: A survey of machine learning platforms. In *15th International Conference on eScience, eScience 2019, San Diego, CA, USA, September 24-27, 2019*, pages 86–95. IEEE, 2019. doi: 10.1109/eScience.2019.00017. URL `https://doi.org/10.1109/eScience.2019.00017`.

Balázs Kégl, Alexandre Boucaud, Mehdi Cherti, Akin Osman Kazakci, Alexandre Gramfort, Guillaume M Lemaitre, Joris Van den Bossche, Djalel Benbouzid, and Camille Marini. The RAMP framework: from reproducibility to transparency in the design and optimization of scientific workflows. International Conference On Machine Learning, July 2018. URL `https://hal.archives-ouvertes.fr/hal-02072341`. Poster.

Divyabh Mishra. Crowdanalytix. 2012. URL `https://www.crowdanalytix.com/`.

Sharada Mohanty, Shivam Khandelwal, and Marcel Salathé. Aicrowd. 2016. URL `https://github.com/AIcrowd/AIcrowd`.

Benedict Neo. 12 data science and ai competitions to advance your skills in 2021. *Towards Data Science*, 2021. URL `https://towardsdatascience.com/12-data-science-ai-competitions-to-advance-your-skills-in-2021-32e3fcb95d8c`.

Humphrey. Quill and David. Penney. *John Harrison, Copley medallist and the L.20,000 longitude prize / by H. Quill ; [line drawings by David Penney].* Antiquarian Horological Society [Ticehurst] ([New House, High St., Ticehurst, Wadhurst, Sussex TN5 7AL]), 1976. ISBN 0901180130.

David Rousseau and Andrey Ustyuzhanin. Machine learning scientific competitions and datasets. 2020. URL `https://arxiv.org/abs/2012.08520`.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

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`.

Deshraj Yadav, Rishabh Jain, Harsh Agrawal, Prithvijit Chattopadhyay, Taranjeet Singh, Akash Jain, Shivkaran Singh, Stefan Lee, and Dhruv Batra. Evalai: Towards better evaluation systems for AI agents. *CoRR*, abs/1902.03570, 2019. URL `http://arxiv.org/abs/1902.03570`.