Renku: a platform for sustainable data science

Rok Roškar[1], **Chandrasekhar Ramakrishnan**[1], **Michele Volpi**[1],
Fernando Perez-Cruz[1], **Mohammad Alisafaee**[2], **Philipp Fischer**[3], **Lilian Gasser**[1],
Eliza Jean Harris[1], **Firat Ozdemir**[1], **Patrick Paitz**[3], **Carl Remlinger**[2],
Luis Salamanca[1], **Ralf Grubenmann**[1], **Tasko Olevski**[1], **Elisabet Capón García**[1],
Lorenzo Cavazzi[1], **Jakub Chrobasik**[2], **Andrea Cordoba**[1], **Alessandro Degano**[2],
Jimena Dupré[1], **Wesley Johnson**[1], **Eike Kettner**[1], **Laura Kinkead**[1],
Seán Murphy[1], **Flora Thiebaut**[1], **Olivier Verscheure**[1,2]
1. Swiss Data Science Center, ETH Zürich, Zürich, Switzerland.
2. Swiss Data Science Center, EPFL, Lausanne, Switzerland.
3. Swiss Federal Institute for Forest, Snow, and Landscape Research, WSL, Birmensdorf, Switzerland

Abstract

Data and code working together is fundamental to machine learning (ML), but the context around datasets and interactions between datasets and code are in general captured only rudimentarily. Context such as how the dataset was prepared and created, what source data were used, what code was used in processing, how the dataset evolved, and where it has been used and reused can provide much insight, but this information is often poorly documented. That is unfortunate since it makes datasets into black-boxes with potentially hidden characteristics that have downstream consequences. We argue that making dataset preparation more accessible and dataset usage easier to record and document would have significant benefits for the ML community: it would allow for greater diversity in datasets by inviting modification to published sources, simplify use of alternative datasets and, in doing so, make results more transparent and robust, while allowing for all contributions to be adequately credited. We present a platform, Renku, designed to support and encourage such sustainable development and use of data, datasets, and code, and we demonstrate its benefits through a few illustrative projects which span the spectrum from dataset creation to dataset consumption and showcasing.

1 Introduction

Datasets are key to all aspects of machine learning (ML) and data science, from the development of novel algorithms to training of production models. Unfortunately, a variety of problems plague the generation, use, and dissemination of datasets [2, 39, 32]. The negative downstream effects of issues related to dataset preparation on model performance and application are well-documented [39]. Less well-documented are the difficulties in understanding and managing the deficiencies on the less technical aspects: the issues of attribution, reuse, dissemination, and replicability. Because ML research is so heavily dependent on datasets, [32] highlight the lacking and insufficiently robust mechanisms for dataset citation and usage tracking as core missing pieces needed to improve the health of the community as a whole. The lack of diverse datasets on which to advance the field has also recently been raised as a point of serious concern [22], and the lack of dataset provenance is a barrier to trustworthy AI systems [19].

Furthermore, beyond requiring planning and design [15], dataset creation requires a significant computational effort, which implies code, pipelines and compute resources. However, in the majority of cases, because dataset creation pipelines are treated as one-offs and are not recorded or published [12], researchers who want to augment a dataset or repeat its construction with different data cannot do so.

37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks.

This significantly limits the engagement of the community in this arguably foundational phase of ML research [44]. The ML community has proposed and adopted some solutions specific to the nature of ML dataset creation and use, such as e.g. *Datasheets for Datasets* [11] (see also [7] for a recent reformulation). While these proposals raise awareness of the importance of datasets and the need for documentation and systematic rigor, they are largely prescriptive, not machine-readable, and cannot easily be automated or validated which makes them quickly become outdated.

Scientific research has struggled with these same issues for many years especially when it comes to valuation of datasets as legitimate research outputs and their citation and attribution. In the past decades, much attention has also been dedicated to the struggles of scientific disciplines to reproduce (sometimes very influential) results [37]. Similarly, the ML community is not immune to the issues of reproducibility [16]; in the case of datasets, where the introduction of hidden biases during the preparatory stages might critically influence the downstream chain of model training and production applications, the need for more comprehensive solutions is high. In this sense, it helps to lean on the concepts espoused by the Open Science movement, where technical questions about robustness, reproducibility and FAIR-ness (FAIR: Findable, Accessible, Interoperable, Reusable as proposed by [43]) overlap with and complement meta concerns about access to knowledge and transparency [8].

To work openly is to structure the research process in such a way that its outcomes can be reproduced, reused, and understood by anyone at any time. This implies structure, discipline, and adherence to norms, which in turn require significant cognitive, technical, monetary and temporal overhead. The *cost* of openness is a significant factor in preventing a wider adoption of these practices [13] despite enormous efforts by research institutions and funding agencies to exert top-down pressure through open science requirements and initiatives.

Openness, however, is not enough. We argue that *sustainability* is the more appropriate mental model, because it implies continuity and dynamics; i.e., simply "opening" a dataset says nothing about how it was assembled and created, where it is eventually used (outside of formal citations), how it might be repackaged, and the many downstream implications that various decisions at each stage of the process might have. The process of dataset creation is for the most part opaque and hidden behind qualitative descriptions in publications. Similarly, sharing source code does not guarantee that anyone can actually run it and, much less, obtain the expected results. Datasets and data are not the final products of the process; they are often deeply intertwined with the models themselves, both in publications and in the general discourse around ML applications. Therefore, the entire cycle must be considered: from dataset assembly and creation to their consumption resulting in eventual deployment of models for benchmarking or in production. While such robust documentation and systematic adherence to standards would be ideal, researchers are pragmatic; they want to get their work done and be recognized for it. In ML research, dataset development tends to be undervalued and is consequently not systematically documented [5]. Furthermore, when voluntary compliance with "best-practices" and efficiency are in competition for time, researchers will tend to optimize for efficiency [3]. In a sustainable system, the actions of individuals translate simultaneously into benefits for themselves (e.g. in the form of increased short-term efficiency or longer-term recognition) and the communities within which they operate (e.g. improved transparency) with little or no overhead.

We present Renku[1], an open-source platform that aims to address some of these problems from another angle: through the creation of a more sustainable system within which ML research is conducted, by incentivising creators and consumers of datasets and models to work in ways that benefit both themselves and their communities. We describe the vision of the Renku platform and its practical implementation; we outline the impact this system can have on dataset development for ML applications; and we share several real-world use cases that demonstrate to varying degrees the practical manifestation of sustainability through the platform, with a specific focus on datasets, their preparation, processing, and dissemination. The use cases illustrate how the platform relieves data scientists and ML researchers from some basic sources of friction present in every data analysis project by: providing easy access to reproducible, containerized environments; simplifying shared data management; providing tooling for building reproducible pipelines; ensuring that data, code, and compute environment are versioned together and always self-consistent. This ecosystem, in turn, allows for the creation and dissemination of datasets, where code and pipelines that are essential for the creation of datasets can readily be inspected, scrutinized, shared, and rerun. The added

[1] `https://renkulab.io`, developed at the Swiss Data Science Center, a strategic focus area of the ETH domain with offices at ETH Zürich, EPFL, and the Paul Scherrer Institute (PSI)

transparency of such a system provides the missing piece to make dataset creation a collaborative, community exercise.

2 Platform vision

The Renku platform's vision is to facilitate sustainable data science. In such a practice, individuals work in a knowledge-capturing way to collaborate, discover, reuse and reproduce results easily.

To motivate this idea, it is helpful to draw a parallel to version control as a tool for *sustainable software development*. Version control contributes to making software development sustainable by helping developers capture and describe changes to the codebase over time. This allows them to return to previous points in time, run the code as it was then, understand motivations for the changes made to the code, and explore implementation alternatives without breaking systems in production. We believe that tools are needed in ML to support similar ways of working: tools that make it possible to return to a project at a previous point in time, understand how the project evolved, and allow for exploration of alternatives while supporting reuse.

The benefits of working in a sustainable data science *environment* are manifold. For example, one can easily return to a project as it was in an earlier state, and still successfully run the code; one can inspect a colleague's project and pick up where they left off; a group lead can quickly identify the uses of a shared dataset and actively scrutinize the results immediately, with zero setup overhead. On the group level, the captured knowledge can serve to facilitate exchange among colleagues, reuse of data and techniques, and to offer members a quick, actionable overview of their group's activities. On the community level, this knowledge capture leads to improved transparency, enhances trust in results, and identifies all artifacts essential to the end-to-end process, such as data pipelines and datasets.

Our vision is, therefore, to build a system that offers simple solutions to problems that data scientists, researchers, and ML users face every day, and in doing so inject just enough structure to make their work more sustainable. The platform can then be used to propagate the information of *what* is being done with data and *how* on a variety of levels and scales. By providing researchers with solutions that alleviate common points of friction (e.g. access to efficient compute and storage), increasing productivity and efficiency, while at the same time capturing knowledge along the whole life-cycle, the Renku platform makes it possible to record the progress of research *as it happens*.

3 Platform design and features

Given the above vision of enabling sustainable data science, the platform is designed around a few basic concepts: (i) an integrated solution for Code, Datasets, Workflows, and Computational environments; (ii) simple interfaces for consistent compute and data access; (iii) building on existing solutions and avoiding vendor lock-in; (iv) connecting everything through a comprehensive, flexible knowledge graph. This section is deliberately brief and omits much detail; further information can be found by visiting the live projects on `https://renkulab.io` (see links in citations).

3.1 Building blocks

Renku makes opinionated choices about a large technology stack to provide the basic building blocks of any data science-related project: Code, Datasets, Workflows, and Computational environments (Figure 1).

Code: Modern data analysis does not happen without code; the problem of working collaboratively with code has largely been solved by the invention of decentralized version control systems like `git`. Renku does not try to reinvent the wheel and relies fully on this industry standard.

Datasets: Renku encourages users to organize their data via *Datasets*, which can be thought of as annotated data containers, i.e. lightweight encapsulations of data, useful for a project together with the relevant metadata. Their true value, however, is that they are an easy way to share and incorporate data into projects either from other Renku projects or external data repositories (e.g. Dataverse or Zenodo), while fully preserving the relevant metadata. If a Dataset is published and receives a DOI, this can be tracked in Renku as well. Datasets can evolve as new data becomes available; this

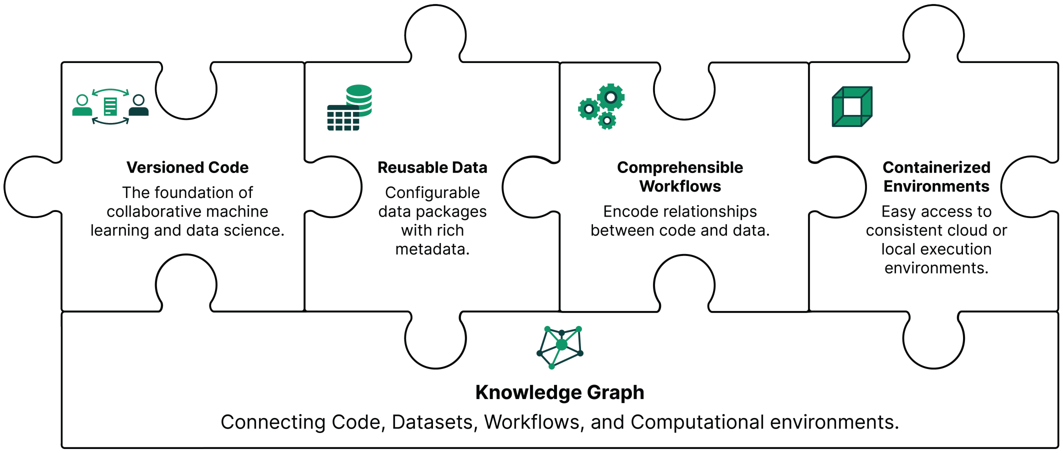
Versioned Code
The foundation of collaborative machine learning and data science.
Reusable Data
Configurable data packages with rich metadata.
Comprehensible Workflows
Encode relationships between code and data.
Containerized Environments
Easy access to consistent cloud or local execution environments.
Knowledge Graph
Connecting Code, Datasets, Workflows, and Computational environments.

Figure 1: A summary of the components of Renku.

provenance is also self-consistently traced in Renku, whether a derived dataset is created by the original author or someone else entirely.

Data can be stored using different backends. The default is to version data together with the code, but store it in git Large File Storage (LFS). Users can also opt to connect a Dataset to an external data source, such as an Amazon Web Services Simple Storage Service (S3) bucket or Azure blob storage[2]. Renku can handle the minutiae of bringing the data to the user's system (including access credential management) so that after the initial setup they no longer have to worry about how to obtain the data. External sources currently do not include databases; while projects may retrieve data from databases through scripts, and store results of queries in files, Datasets cannot currently use databases as data sources directly.

Renku supports ways of restricting access to data. For data stored in git LFS within a Renku instance, the data access is limited to those who have access to the repository. If data is external, access is managed by a third-party and not by Renku. The infrastructure that hosts the compute resources for running workflows and interactive sessions is designed to be multi-tenant and prevents users from accessing resources allocated to others. While this should be sufficient for many kinds of sensitive data, some highly-sensitive or confidential data may require additional precautions on the data at rest, like encryption, which Renku does not itself provide. In those cases, we recommend that a Renku instance is deployed in a secure environment and urge users to consider their levels of acceptable risk to avoid potential leakage of sensitive data. More information about deployment and administration can be found in Section 3 of the Supplementary Materials.

Workflows: Workflows are where code meets data; a workflow can be anything from a shell script to a full-blown distributed pipeline. Renku provides a very simple workflow system which captures the basic relationship between data and any code that can be executed on the command line. The primary function of Renku workflows is to capture provenance and make it possible to reproduce and rerun computational pipelines involving several intertwined steps that would normally run as separate scripts, or even as a part of one (usually very long) script. The system aims to relieve the researcher from the burden of remembering how a specific result was generated.

Once a workflow is executed, content hashes of all components are recorded in the metadata and can be checked against any future state. If any of the pieces are updated (code or data), Renku can generate an execution plan to rerun the parts that need to be updated. The workflows are recorded using an open ontology based on W3C recommendations[3] and *schema.org* but can be serialized into a workflow language to execute on a variety of backends including cloud and HPC. Currently, we

[2] Integrations are not limited to these services and additional backends are continously being added.
[3] `https://www.w3.org/TR/prov-o/`

support the *Common Workflow Language*[6], *toil*, and *Argo* workflows[4], but others can be added by external extensions via a simple plug-in system.

Computational environment: Code requires a computational environment in which to run; this environment includes software libraries and languages, compilers, system libraries, and the operating system. Failure to encapsulate this environment to a sufficient degree is the primary hurdle for collaboration, reuse and reproducibility in computational research [23]. Renku makes heavy use of container technologies to alleviate some of the pain related to computational environments. By default, each project includes a `Dockerfile` that builds a docker image for every commit in the project's repository. This ensures that a computational environment matching any commit of the project is always available. Users can either use these images on their local machine (or another compute resource) or launch hosted interactive sessions directly in RenkuLab, the hosted part of the platform.

3.2 Interfaces for consistent compute and data access

The Renku platform consists of a RenkuLab instance (e.g. our public instance at `https://renkulab.io`) with a browser-based interface and a Python library, which also provides a command-line interface (CLI)[5]. The two interfaces can be used together or independently, depending on the project constraints. We strive for feature parity between the two interfaces, meaning that users should have a consistent experience with launching compute sessions and accessing data in both cases. The CLI in addition allows users to take a project completely off-line to another compute resource, without relying on any hosted (cloud) infrastructure. This is particularly useful in cases involving sensitive data or restricted environments.

3.3 Building on Open Source

Renku is open source[6] and licensed under the Apache 2.0 License. It builds and relies on a number of well-known open-source technologies: Kubernetes at the infrastructure layer, docker for containerization, git and GitLab for version control and repository management, and the Jupyter ecosystem [21] for interactive sessions. The platform is designed to prevent vendor lock-in, i.e. users are not obliged to keep working on their project within the hosted platform or even using any Renku tooling. They may simply take their Project (a git repository) anywhere and use it with standard tools like git and docker.

3.4 Connect Everything: the Renku Knowledge Graph

A central component of the Renku platform is the knowledge graph (KG) made up of metadata aggregated from all the projects on the platform. All primary Renku entities (Projects, Code, Datasets, Workflows) are represented in the KG. This metadata can facilitate many different kinds of functionality. For example, recording a workflow allows a user to use the system to re-execute parts of their computation if some upstream data changes. Conversely, if a user is interested in a particular Dataset, they can use the KG to understand how it has been created or analyzed in the past, who has experience with the data and what kinds of results have been derived from it. The KG is based on RDF and designed to be interoperable and extensible by 3rd party plug-ins with domain-specific metadata to facilitate use cases across a broad spectrum of applications. One example is a plug-in that automatically adds MLSchema [33] annotations to Workflows[7].

The metadata stored in the KG is subject to the same access control restrictions as code and data. A user only sees the metadata associated with Projects or Datasets that they are otherwise allowed to access, including when performing a global search. For more details on the KG, please see Section 1 of the Supplementary Materials.

[4] At the time of writing, with limited functionality.

[5] `https://github.com/SwissDataScienceCenter/renku-python`

[6] `https://github.com/SwissDataScienceCenter/renku`

[7] `https://github.com/SwissDataScienceCenter/renku-mls`

4 Renku for Community Dataset Development

We believe that the feature-set of Renku and its design principles make it a great platform for the development, dissemination and sharing of ML datasets and benchmarks. We envision that Renku will be used to enable new forms of collaboration on community dataset development and to facilitate the extension and reuse of different datasets and benchmarks based on reusable code and pipelines.

4.1 Transparency and trust

Responsible, sustainable development of datasets for use in the ML research community requires transparency. A dataset author, for example, desires that the downstream use of their data can quickly and easily be discovered. Conversely, a dataset user or consumer needs to be able to quickly examine the data and to understand its provenance to be absolutely certain about any assumptions, tweaks or modifications that may impact downstream interpretation. A data consumer may also like to discover uses of the data they are interested in, to gain inspiration and build on existing work easily.

In Renku, both of these points of view are satisfied through the integration of the four key platform elements (Code, Dataset, Workflows and Compute environments) with the metadata layer, the Renku knowledge graph. A Dataset author, for example, can use Renku tools to track how a dataset is prepared from raw data to the final state that is distributed for wider use. The pipeline that generates the final Dataset is self-consistently recorded and may be re-executed in the controlled compute environment to generate new datasets with different assumptions or to verify that the creation of the dataset itself is repeatable and reproducible. The ability to quickly work hands-on with a dataset with a pre-configured compute environment and executable code examples greatly simplifies the discovery and vetting process, inspiring trust and stimulating reuse and collaboration.

4.2 Dissemination and sharing

Dataset authors aspire to make their work easily accessible to the widest possible audience and applicable to a wide range of use cases. Dataset users, on the other hand wish to quickly discover datasets relevant to their research interests and use them with minimal friction. Renku provides authors with the means to package and distribute their Datasets easily and embed them into reproducible computational environments with sample code to demonstrate and document use. Documentation in this case goes beyond text descriptions, but can include code examples, visualizations or dashboards. This allows authors to build demonstrations of the dataset's application directly in their project; these can be open to the public and provisioned quickly.

Once a user finds a dataset, they can add it to their own Renku project with a click. In doing so, the user does not need to worry about how to access (download) the data; the platform makes sure that the Dataset is accessible wherever the project is used. Data access information, whether the data is stored within the repository or in a cloud-storage location, is preserved with the project ensuring that the data is made available either in hosted interactive sessions or locally with minimal friction.

By making it easy for anyone to provide datasets for broader consumption in a frictionless way, Renku makes it possible to expedite the dissemination of datasets within the community. Currently, a handful of datasets are used by an overwhelmingly large percentage of papers [22]; not only because they are useful datasets, but also because they are made easily available, coupled with user-friendly tools for data handling, and demo projects and examples (e.g. the COCO dataset [25]). With Renku, it is straightforward to make the dataset available along with all the code needed to (re)create it or modify it, bundled with a compute environment. This opens the door to more robustly inspect published datasets, and to use them as a starting point for generating new, complementary ones.

4.3 Identifying use and impact

By using a dataset within the platform, the information about the use is preserved within the knowledge graph. This makes it possible for dataset authors or members of the wider community to trace the usage of the data and self-consistently identify datasets that have the most traction within the community. The knowledge graph enables evaluation beyond the superficial "number of downloads" or "views"; instead, we can analyze the network of downstream applications of a Dataset, e.g. whether they lead to results and publications, or whether they have an impact across many

Platform	Versioning	Environments	Workflows	Datasets	Apps	Language
DATTSFLOW [31]	+	+	+	+	-	Python
KG-FRUS [38]	+	+	+	+	-	Python
MLTox [10]	+	+	+	+	-	Python
TimeFRAME [9]	+	+	+	-	+	R
MLED [35]	+	+	-	-	-	Python
OADAT [29]	+	+	-	-	-	Python

Table 1: Summary of Renku features showcased in the example use case projects. These features are derived from the **building blocks** mentioned above, with versioning coming from the *Code* building block. The *App* feature allows deployment of project-specific UIs.

different institutions of the research community. This type of analysis and interpretation of dataset impact is not currently possible in other platforms made for data sharing. The same information can be used more specifically for community benchmarking purposes, as the *Omnibenchmark* project [26] demonstrates.

Apart from adding dimensions to assessing dataset applications, the ability to quickly identify dataset use *in progress* can have significant impact on collaboration within a community. The usual academic cycle of publication means that pieces of the research process that are fundamental to its success (e.g. datasets and benchmarks) are not recognized until the publications that use them are complete. This means that information only properly proliferates with a significant delay. Instead, in a framework like Renku, information exchange and attribution could take place much sooner.

5 Example use cases

We have collected a number of use cases from real-world projects conducted at the Swiss Data Science Center and the ETH domain, which illustrate the value of the Renku platform for collaborative dataset development, use, and reuse. The projects are related to the development and application of ML in support of scientific research. In all cases, the challenge is to combine domain knowledge of the research question with specific requirements of downstream ML work. The case studies presented here bridge the gap from domain to data science and provide experts from both sides with the means to examine, confirm, and extend the datasets and their analyses, without extensive investments of time and infrastructure. Links to all projects are included in the citations. Further details about the TimeFRAME and MLED projects can be found in Section 4 of the Supplementary Materials.

It is important to note that (at the time of writing) not all of the sample projects utilize all of Renku's functionality and that this is perfectly acceptable - depending on the stage of the project, researchers have different priorities to juggle. The summary of the adoption of various Renku features by the example projects can be found in Table 1. The adoption curve of the platform's features is intended to be gradual - once a project is utilizing the platform, the "missing pieces" (e.g. explicit workflows or data encapsulation through Datasets) can happen at any point when results start to take shape.

DATSSFLOW: ML-driven seismic signal classification The DATSSFLOW case study[31] deals with developing automated data parsing pipelines and automatic retraining of feature extraction models in a geophysical context. The project utilizes a multi-step pipeline, expressed as a set of Renku workflow steps, to convert publicly available raw seismic data into a series of Renku Datasets that may be reused in downstream tasks and applications. These steps are shown schematically in the right panel of Figure 2; the manifestation of the workflow, as seen via the KG is shown in the right panel.

All of the steps above are linked together: as soon as new data comes in, the entire pipeline is automatically re-executed and the feature extraction model updated. Renku significantly reduces the burden of manually running each and every script for each single dataset available, for current datasets but also from those coming from future campaigns. Data from more stations available from the International Federation of Digital Seismograph Networks (FDSN) could easily be added to the first Renku Workflow to adapt the entire pipeline to a different application. Each step is documented via instructions and READMEs, along with code, parametrizations and execution Workflows, enabling automatic updates to the full data pipeline, even for people not familiar with the project.

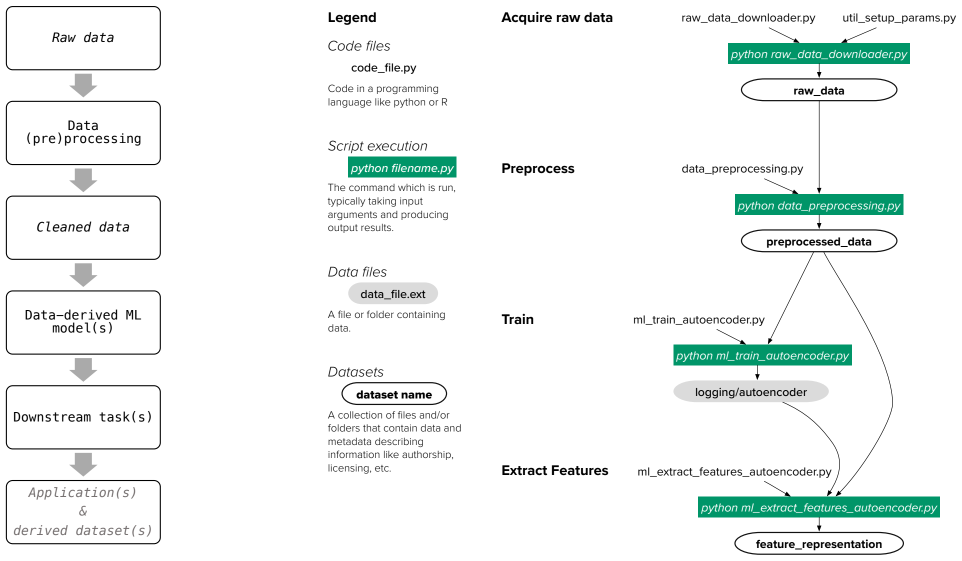
Raw data
Data (pre)processing
Cleaned data
Data-derived ML model(s)
Downstream task(s)
Application(s) & derived dataset(s)
Legend
Code files
code_file.py
Code in a programming language like python or R
Script execution
python filename.py
The command which is run, typically taking input arguments and producing output results.
Data files
data_file.ext
A file or folder containing data.
Datasets
dataset name
A collection of files and/or folders that contain data and metadata describing information like authorship, licensing, etc.
Acquire raw data
raw_data_downloader.py
util_setup_params.py
python raw_data_downloader.py
raw_data
Preprocess
data_preprocessing.py
python data_preprocessing.py
preprocessed_data
Train
ml_train_autoencoder.py
python ml_train_autoencoder.py
logging/autoencoder
Extract Features
ml_extract_features_autoencoder.py
python ml_extract_features_autoencoder.py
feature_representation

Figure 2: DATSSFLOW conceptual data flow (left) and the automatically generated knowledge graph workflow representation (right).

KG-FRUS Renku has been utilized to track the creation of the KG-FRUS dataset[38], a graph-based dataset built from the diplomatic documents of the Foreign Relations of the United States (FRUS) office. In [45], the authors advocate that a knowledge graph (KG) is the representation required to approach the study of complex fields such as diplomacy and political sciences, where synergies and interactions between international actors are extremely relevant, and which are hitherto ignored in standard corpora. As this dataset is based on public data dealing with public policy and world events, it is important to be transparent about its creation. Utilizing a system like Renku means that anyone can check, reuse, alter and rerun the pipeline to regenerate and check the consistency of the dataset, or even to produce their own dataset from similar data.

MLTox: Enhancing toxicological testing through machine learning Within the MLTox project, focused on providing ML-based alternatives to animal testing for ecotoxicological studies, the authors generated a benchmark dataset, ADORE, on acute mortality of three taxonomic groups (fish, crustaceans and algae) [40]. The core of ADORE originates from the ECOTOX database [28], which can be downloaded as a set of `.txt` files. The relevant files were harmonized and pre-filtered to only contain entries suitable to model acute toxicity in fish, crustaceans and algae. We expanded the core dataset with taxonomic and chemical information. The data was then filtered to only contain entries on acute mortality, which is defined differently for the three taxonomic groups. Additionally, we only retain data points for which information from all sources is available. The filtered dataset mainly contains entries from organic molecules, representing a rather small chemical space, which have been tested on species with taxonomic information available.

In general, and for this dataset specifically, data splitting has a substantial impact on the modeling output [20]. We therefore proposed specific challenges on subsets of the data, included the corresponding data subsets with splittings as well as in-depth characterization and discussion of train-test splitting approaches [40]. The entire dataset, code, and pipeline needed to generate it are available on Renku [10]. Researchers interested in using or re-generating the dataset can easily do so within the RenkuLab environment. We envision the ADORE dataset being used to accelerate research in ecotoxicity to move towards a reduction and finally a replacement of animal testing.

OADAT: Large scale optoacoustic imaging data OADAT is one of the outcomes of a larger project, which proposed several methodologies to improve image quality by means of geometric distortion correction, noise reduction, and semantic segmentation, for a novel emerging non-invasive clinical imaging modality called *optoacoustic imaging*. Beyond the methodologies, the project also

Platform	Versioning	Datasets	Environments	Provenance	Apps	Languages
Renku	+	+	+	+	+	Multiple
Whole Tale	+	+	+	+	-	Multiple
Hugging Face	+	+	+	-	+	Multiple
Kaggle	+	+	+	-	-	Multiple
DVC	+	+	-	+	-	N/A
Google Colab	+	-	+	-	-	Python
OpenML	-	+	-	+	-	Multiple
Papers with Code	-	+	-	-	-	N/A
Binder	-	-	+	-	-	Multiple

Table 2: Summary of platforms and tools overlapping with Renku features focusing on support for **version** control, **datasets**, reproducible **environments**, data **provenance**, GUI **apps**, and programming **languages**

procured a first of its kind large scale optoacoustics dataset consisting of experimental and simulated samples collected using varying imaging devices, conditions, and volunteers.

[30] reference the Renku project[29], where one can not only access each component in the aforementioned workflow, but can also customize individual components to (i) evaluate pretrained models on a different (proprietary) optoacoustic dataset, (ii) train the models on the datasets, (iii) evaluate performance of pretrained models, and even (iv) test the pretrained models on proprietary optoacoustic images qualitatively to see if they can be integrated into other workflows of domain science (in this case optoacoustic imaging) experts. Each step is designed to accept other models or datasets for efficient comparison and evaluation.

6 Comparison to other systems

Renku provides solutions to problems that are well known and widely acknowledged issues in ML and data science more generally: there are many tools and platforms that overlap with Renku in addressing them. To provide a better understanding of where Renku fits in the landscape, we give a brief comparison with some of the most relevant platforms: Whole Tale [4], Hugging Face[14], Kaggle[17], DVC[18], Google Colab[36], OpenML [41], Papers with Code[1], and Binder[34]. A summary is provided in Table 2.

We want to emphasize that we do not see Renku as being in competition with any of these platforms. Each one targets a different set of usage scenarios, making it difficult to do a direct comparison in a fair way. As we argue above, we see working sustainably as imperative to the development of machine learning and encourage researchers to use whatever tools they find helpful in achieving this goal. In the comparison here, the focus is on features, but other aspects like *can I deploy my own instance*, *is paid support available*, *is the platform operated for profit*, *can I access GPUs* may certainly be relevant.

Renku shares the most in common with the first three of the aforementioned platforms: Whole Tale, Hugging Face, and Kaggle. Like Renku, all these platforms support versioning code, defining datasets, and creating reproducible environments in the cloud. Of these platforms, Whole Tale and Renku are the two that support finer-grained tracking of data provenance (referred to as *Workflows* in Renku). Hugging Face and Renku let users deploy GUI-based apps that let others explore a project interactively without having to write or execute code.

Two projects that overlap in a smaller subset of features with Renku are DVC and Google Colab. DVC is a tool for *D*ata *V*ersion *C*ontrol, and supports tracking versions of data, models, and provenance of results, features that Renku also supports. Like Renku, Hugging Face, and Kaggle, Google Colab offers cloud-based environments with optional GPU access. Colab is, however, tied to notebook-based Python environments and does not provide support for datasets.

The final three projects are the most orthogonal to Renku among this list. OpenML is a library and platform for accessing datasets and publishing detailed provenance information about how analysis was performed. Its feature set complements Renku very nicely, since Renku does not offer the same level of support for retrieving datasets and tasks or detail about how ML pipelines are built, but

Renku provides reproducible cloud-based environments and the ability to track provenance about other aspects of projects (like data pre-processing). Papers with Code is a repository of publications, and code and datasets that go along with them. Publications are not a central focus of Renku. Finally, Binder is a tool that very conveniently allows creating a cloud environment for running notebooks in Python, R, and Julia, but it is not designed to create environments for the development of ML projects.

7 Current status, Limitations and Future work

As of this writing, the Renku platform runs in at least seven different instances, including two at universities in Switzerland and one used by the Swiss Federal Statistical Office. The public instance is open to all and can be found at `https://renkulab.io`. It currently counts around 8'000 users. Apart from ML projects, some of which we highlighted above, Renku users span a whole range of scientific domains (e.g. hydrology [27, 24], climate science [42], and bioinformatics [26]), experience levels, and IT literacy. Many courses are taught on the platform, taking advantage of the easy set up of consistent, reusable compute environments.

The current version of the platform has focused on the usability of the CLI and the web app, the generation of the underlying metadata and its handling in production and the reusable compute environments, which are the most heavily used feature. Currently, the platform lacks the functionality for users to generate views of the knowledge graph easily (Section 3.4), although the metadata to enable such views already exists. Therefore, cross-project views or more complex queries relevant for discovering Datasets and their impact are not easily accessible. We are working on features that will enable such exploration.

One of the main limitations of Datasets is the reliance on git-LFS as the data storage backend (Section 3.1). It is inflexible, prone to error, and does not work well for large amounts of data. In our efforts to make the platform a landing zone for all types of data science and ML projects, implementation work is already in motion to simplify and extend the Dataset functionality; on the one hand to make it easier to connect to other sources of data, and on the other to interface with data repositories and archives for preservation and minting of DOIs. Datasets also currently lack more extensive metadata, such as licensing or publication information, which will be added in future iterations.

A common problem is availability and flexibility of compute resources, as users are currently limited to using the on-line compute resources provided by the Renku instance they are using (Section 3.1). To better connect users to their data and compute resources, we plan to enable users to connect more seamlessly to other providers like public clouds or HPC clusters. At the same time, interactive computing platforms are prone to not utilizing compute resources well and we strive to implement features that help users avoid wasteful consumption.

We sincerely hope that the broader ML community will consider Renku as a platform for building, developing, and ultimately using and disseminating datasets. We look forward to engaging with questions, issues, and feature requests in our public channels[8].

References

[1] Meta AI. Papers with Code `https://paperswithcode.com`, 2023.

[2] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA, 2021. Association for Computing Machinery.

[3] Barry Bozeman, Jan Youtie, and Jiwon Jung. Death by a thousand 10-minute tasks: Workarounds and noncompliance in university research administration. *Administration & Society*, 53(4):527–568, 2021.

[8] Discourse forum at `https://renku.discourse.group`, main GitHub repository at `https://github.com/SwissDataScienceCenter/renku`

[4] Adam Brinckman, Kyle Chard, Niall Gaffney, Mihael Hategan, Matthew B Jones, Kacper Kowalik, Sivakumar Kulasekaran, Bertram Ludäscher, Bryce D Mecum, Jarek Nabrzyski, et al. Computing environments for reproducibility: Capturing the "whole tale". *Future Generation Computer Systems*, 94:854–867, 2019.

[5] Inha Cha, Juhyun Oh, Cheul Young Park, Jiyoon Han, and Hwalsuk Lee. Unlocking the tacit knowledge of data work in machine learning. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI EA '23, New York, NY, USA, 2023. Association for Computing Machinery.

[6] Michael R. Crusoe, Sanne Abeln, Alexandru Iosup, Peter Amstutz, John Chilton, Nebojsa Tijanic, Hervé Ménager, Stian Soiland-Reyes, and Carole A. Goble. Methods included: Standardizing computational reuse and portability with the common workflow language. *CoRR*, abs/2105.07028, 2021.

[7] Andy Donald, Apostolos Galanopoulos, Edward Curry, Emir Muñoz, Ihsan Ullah, M. A. Waskow, Maciej Dabrowski, and Manan Kalra. Towards a semantic approach for linked dataspace, model and data cards. In *Companion Proceedings of the ACM Web Conference 2023*, WWW '23 Companion, page 1468–1473, New York, NY, USA, 2023. Association for Computing Machinery.

[8] Benedikt Fecher and Sascha Friesike. *Open Science: One Term, Five Schools of Thought*, pages 17–47. Springer International Publishing, Cham, 2014.

[9] Philipp Fischer. Renku project: N2O Pathway Analysis `https://renkulab.io/projects/fischphi/n2o-pathway-analysis`, 2023.

[10] Lili Gasser. Renku project: ADORE `https://renkulab.io/projects/mltox/adore`, 2023.

[11] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna M. Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *CoRR*, abs/1803.09010, 2018.

[12] Amy K. Heger, Liz B. Marquis, Mihaela Vorvoreanu, Hanna Wallach, and Jennifer Wortman Vaughan. Understanding machine learning practitioners' data documentation perceptions, needs, challenges, and desiderata. *Proc. ACM Hum.-Comput. Interact.*, 6(CSCW2), nov 2022.

[13] Thomas J. Hostler. The Invisible Workload of Open Research. *Journal of Trial & Error*, may 4 2023. https://journal.trialanderror.org/pub/the-invisible-workload.

[14] Inc. Hugging Face. Hugging Face `https://huggingface.co`, 2023.

[15] Ben Hutchinson, Andrew Smart, Alex Hanna, Emily Denton, Christina Greer, Oddur Kjartansson, Parker Barnes, and Margaret Mitchell. Towards accountability for machine learning datasets: Practices from software engineering and infrastructure. *CoRR*, abs/2010.13561, 2020.

[16] Matthew Hutson. Artificial intelligence faces reproducibility crisis. *Science*, 359(6377):725–726, 2018.

[17] Kaggle Inc. Kaggle `https://www.kaggle.com`, 2023.

[18] Inc. Iterative. DVC `https://dvc.org`, 2023.

[19] Amruta Kale, Tin Nguyen, Jr. Harris, Frederick C., Chenhao Li, Jiyin Zhang, and Xiaogang Ma. Provenance documentation to enable explainable and trustworthy AI: A literature review. *Data Intelligence*, 5(1):139–162, 03 2023.

[20] Sayash Kapoor and Arvind Narayanan. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, page 100804, 2023.

[21] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, and Carol Willing. Jupyter notebooks – a publishing format for reproducible computational workflows. In F. Loizides and B. Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87 – 90. IOS Press, 2016.

[22] Bernard Koch, Emily Denton, Alex Hanna, and Jacob Gates Foster. Reduced, reused and recycled: The life of a dataset in machine learning research. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[23] M. S. Krafczyk, A. Shi, A. Bhaskar, D. Marinov, and V. Stodden. Learning from reproducing computational results: Introducing three principles and the reproduction package. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2197), May 2021.

[24] Louis Krieger, Remko Nijzink, Gitanjali Thakur, Chandrasekhar Ramakrishnan, Rok Roškar, and Stan Schymanski. Repeatable and reproducible workflows using the RENKU open science platform. In *EGU General Assembly Conference Abstracts*, EGU General Assembly Conference Abstracts, pages EGU21–7655, April 2021.

[25] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[26] Almut Lütge, Anthony Sonrel, Izaskun Mallona, Charlotte Soneson, Federico Marini, Søren Helweg Dam, Oksana Riba Grognuz, Rok Roškar, and Mark D Robinson. "omnibenchmark: open continuous collaborative benchmarking of computational biology methods". *To be available on bioarxiv upon submission*, 2023.

[27] Remko Nijzink, Chandrasekhar Ramakrishnan, Rok Roškar, and Stan Schymanski. A repeatable and reproducible modelling workflow using the Vegetation Optimality Model and RENKU. In *EGU General Assembly Conference Abstracts*, EGU General Assembly Conference Abstracts, page 9228, May 2020.

[28] Jennifer H. Olker, Colleen M. Elonen, Anne Pilli, Arne Anderson, Brian Kinziger, Stephen Erickson, Michael Skopinski, Anita Pomplun, Carlie A. LaLone, Christine L. Russom, and Dale Hoff. The ECOTOXicology Knowledgebase: A Curated Database of Ecologically Relevant Toxicity Tests to Support Environmental Research and Risk Assessment. *Environmental Toxicology and Chemistry*, 41(6):1520–1539, 2022. 17 citations (Crossref) [2023-05-23] _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/etc.5324.

[29] Firat Ozdemir. Renku project: OADAT `https://renkulab.io/projects/firat.ozdemir/oadat-evaluate`, 2023.

[30] Firat Ozdemir, Berkan Lafci, Xose Luis Dean-Ben, Daniel Razansky, and Fernando Perez-Cruz. OADAT: Experimental and synthetic clinical optoacoustic data for standardized image processing. *Transactions on Machine Learning Research*, 2023.

[31] Patrick Paitz, Michele Volpi, and Francois Kamper. Renku project: DATSSFLOW `https://renkulab.io/projects/patrick.paitz/neurips-renku-seismic-example`, 2023.

[32] Kenneth Peng, Arunesh Mathur, and Arvind Narayanan. Mitigating dataset harms requires stewardship: Lessons from 1000 papers. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran, 2021.

[33] Gustavo Correa Publio, Diego Esteves, Agnieszka Lawrynowicz, Pance Panov, Larisa N. Soldatova, Tommaso Soru, Joaquin Vanschoren, and Hamid Zafar. Ml-schema: Exposing the semantics of machine learning with schemas and ontologies. *CoRR*, abs/1807.05351, 2018.

[34] Benjamin Ragan-Kelley, Carol Willing, F Akici, D Lippa, D Niederhut, and M Pacer. Binder 2.0-reproducible, interactive, sharable environments for science at scale. In *Proceedings of the 17th python in science conference*, pages 113–120. F. Akici, D. Lippa, D. Niederhut, and M. Pacer, eds., 2018.

[35] Carl Remlinger. Renku project: MLED `https://renkulab.io/projects/carl.remlinger/mled`, 2023.

[36] Google Research. Google Colab `https://colab.research.google.com`, 2023.

[37] Peter Rodgers and Andy Collings. Reproducibility in cancer biology: What have we learned? *eLife*, 10:e75830, dec 2021.

[38] Luis Salamanca. Renku project: KG-FRUS `https://renkulab.io/projects/luis.salamanca/kg-frus`, 2023.

[39] Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen K. Paritosh, and Lora Aroyo. "everyone wants to do the model work, not the data work": Data cascades in high-stakes AI. In Yoshifumi Kitamura, Aaron Quigley, Katherine Isbister, Takeo Igarashi, Pernille Bjørn, and Steven Mark Drucker, editors, *CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama, Japan, May 8-13, 2021*, pages 39:1–39:15. ACM, 2021.

[40] Christoph Schür, Lilian Gasser, Fernando Perez-Cruz, Kristin Schirmer, and Marco Baity-Jesi. ADORE: A Benchmark Dataset for Machine Learning in Ecotoxicology. preprint, Bioinformatics, May 2023.

[41] Jan N Van Rijn, Bernd Bischl, Luis Torgo, Bo Gao, Venkatesh Umaashankar, Simon Fischer, Patrick Winter, Bernd Wiswedel, Michael R Berthold, and Joaquin Vanschoren. Openml: A collaborative science platform. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*, pages 645–649. Springer, 2013.

[42] F. Vitart, A. W. Robertson, A. Spring, F. Pinault, R. Roškar, W. Cao, S. Bech, A. Bienkowski, N. Caltabiano, E. De Coning, B. Denis, A. Dirkson, J. Dramsch, P. Dueben, J. Gierschendorf, H. S. Kim, K. Nowak, D. Landry, L. Lledó, L. Palma, S. Rasp, and S. Zhou. Outcomes of the WMO Prize Challenge to Improve Subseasonal to Seasonal Predictions Using Artificial Intelligence. *Bulletin of the American Meteorological Society*, 103(12):E2878–E2886, December 2022.

[43] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3, 2016.

[44] Amy X Zhang, Michael Muller, and Dakuo Wang. How do data science workers collaborate? roles, workflows, and tools. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW1):1–23, 2020.

[45] Gökberk Özsoy, Luis Salamanca, Matthew Connelly, Raymond Hicks, and Fernando Pérez-Cruz. KG-FRUS: a Novel Graph-based Dataset of 127 years of US Diplomatic Relations. *Submitted to NeurIPS - Datasets and benchmarks track*, 2023.