Empirical Software Engineering manuscript No.
(will be inserted by the editor)

On the Suitability of Hugging Face Hub for Empirical Studies

Adem Ait · Javier Luis Cánovas Izquierdo · Jordi Cabot

the date of receipt and acceptance should be inserted later

Abstract CONTEXT. Empirical studies in software engineering mainly rely on the data available on code-hosting platforms, being GITHUB the most representative. Nevertheless, in the last years, the emergence of Machine Learning (ML) has led to the development of platforms specifically designed for hosting ML-based projects, with HUGGING FACE HUB (HFH) as the most popular one. So far, there have been no studies evaluating the potential of HFH for such studies.
OBJECTIVE. We aim at performing an exploratory study of the current state of HFH and its suitability to be used as a source platform for empirical studies.
METHOD. We conduct a qualitative and quantitative analysis of HFH. The former will be performed by comparing the features of HFH with those of other code-hosting platforms, such as GITHUB and GITLAB. The latter will be performed by analyzing the data available in HFH.
RESULTS. We propose a feature framework to characterize HFH and report on the current usage of the platform, both in terms of number and types of projects (and surrounding community) and the features they mostly rely on.
CONCLUSIONS. The results confirm that HFH offers enough features and diverse enough data to be the source of relevant empirical studies on the development, evolution and usage of AI-related projects. The results also triggered

A. Ait
University of Luxembourg
Esch-sur-Alzette, Luxembourg
E-mail: adem.ait@uni.lu

J.L. Cánovas Izquierdo
IN3 – UOC
Barcelona, Spain
E-mail: jcanovasi@uoc.edu

J. Cabot
Luxembourg Institute of Science and Technology, University of Luxembourg
Esch-sur-Alzette, Luxembourg
E-mail: jordi.cabot@list.lu

a discussion on aspects of HFH that should be considered when performing such empirical studies.

Keywords Mining Software Repositories · Data Analysis · Empirical Study · ML · Hugging Face Hub

1 Introduction

The development of empirical studies in Open-Source Software (OSS) requires large amounts of data regarding software development events and developer actions, which are typically collected from code-hosting platforms. Code-hosting platforms (also referred simply as platforms, from now on) are built on top of a version control system, such as Git, and provide collaboration tools such as issue trackers, discussions, and wikis; as well as social features such as the possibility to watch, follow and like other users and projects. Among them, GitHub has emerged as the largest code-hosting site in the world, with more than 80 million users and 200 million repositories.

The emergence of Machine Learning (ML) has led to the development of platforms specifically designed for developing and hosting ML-based projects, being Hugging Face Hub (HFH) the most popular one. In HFH, developers can publish and share their ML-based projects, as well as reuse datasets, pre-trained models and other ML artifacts. As of March 2024, the platform hosts more than 600k public repositories, and this number is growing fast.

In the last months, HFH has been evolving and incorporating new features typically found in other code-hosting platforms. For instance, the ability to create discussions or submit change requests via a mechanism similar to pull requests. Thus, enabling more complex interactions and development workflows within the platform. This evolution, its growing popularity and its ML-specific features, such as the integration of the hosted models in the Transformers library, make HFH a promising source of data for empirical studies. Despite this, the current status of the platform may involve relevant perils that could hinder its use in this type of studies.

In this sense, this paper aims to analyze the current state of HFH and determine its suitability to be used as a source for empirical studies. We define as suitability the ability to use HFH's features and data to perform interesting empirical studies akin to (in terms of relevance and complexity) those we typically see in other platforms. This implies evaluating whether HFH offers enough features and enough data (both in terms of size and diversity) for such empirical studies.

To this aim, we propose a systematic method to characterize the set of features provided by HFH and then study the availability and quality of the data available in HFH. Later, we discuss the results to evaluate their impact in different scenarios commonly found in empirical studies.

The rest of the paper is structured as follows. Sections 2 and 3 provide the background and related work, respectively. Section 4 presents the methodology followed in our study. Section 5 describe the results. Sections 6 and 7 presents the discussion points and the threats to validity, respectively. Finally, Section 8 concludes the paper.

2 Background

Hugging Face (HF), the company behind HFH, is an AI company originally known for its Natural Language Processing (NLP) model called Hierarchical Multi-Task Learning (HMTL) (Sanh et al., 2019) or for the Transformers library (Wolf et al., 2020), which provides APIs and tools to easily download and train state-of-the-art pretrained models.

Nevertheless, it became a household name thanks to the creation of HFH, its ML-based hosting platform, with the goal of building the largest open-source collection of ML artifacts (also referred to as projects in the context of this paper) to advance and democratize the access to ML for everyone. HFH is a Git-based online code-hosting platform aimed at providing a hosting site for all kinds of ML artifacts, namely: (1) models, pretrained models that can be used with the Transformers library; (2) datasets, which can be used to train ML models; and (3) spaces, demo apps to showcase ML models.

The storage for these artifacts relies on Git repositories, where each repository is presented on the HFH website via three tabs, namely: card, files and community. Table 1 shows a detailed list of the contents of each tab per repository type. The card is the front face of the repository, and it is different for each repository type. For model repositories, HFH provides a guide to fill the `README.md` file in order to ensure that users report all available metadata. The metadata that should be reported is the model description, its intended uses and potential limitations as detailed in Mitchell et al. (2019), the training parameters, among others. To facilitate this task, HFH provides a template to help report all the possible fields.[1] Besides the metadata, there are also interfaces to perform inferences to the model using the Inference API[2], and to visualize and download the tensors stored in the `.safetensors` file. Furthermore, there is a list of dataset and spaces dependencies, and the model tree, which contains the fine-tunes, adapters, merges, and quantizations of a base model.[3] For both models and datasets there is an interface to use the model or dataset with its corresponding libraries (e.g., Transformers, CroissantML, etc.). Regarding dataset repositories, the metadata is composed of license, language, and size, among others. However, HFH also reports the size of the files in the repository, along with the equivalent size of the auto-converted Parquet files,[4] and the number of rows. Furthermore, it shows the dependencies and downloads as in model repositories. The card for space repositories is the most different and changes from one space to another, as it is designed to provide a demo of an ML model. Next to the repository card, the file tab displays

[1] `https://huggingface.co/docs/hub/model-card-annotated`
[2] `https://huggingface.co/docs/api-inference`
[3] `https://huggingface.co/docs/hub/model-cards#specifying-a-base-model`
[4] `https://huggingface.co/docs/dataset-viewer/en/parquet`

Table 1: Contents of the repository tabs in HFH.

Tab	Repository Type	Contents	Description
Card	Model	ReadME	Metadata of model
		Downloads	Downloads of last month
		Usage interface	Integration with several libraries to use the dataset
		Inference API	Interface for model inferences
		Safetensors interface	Interface to download and visualize the `.safetensors` file
		Repository dependencies	Model tree, list of datasets used to train and spaces using the model
		collections	Collections that contain the model
	Dataset	ReadME	Metadata of dataset
		Dataset viewer	Interface to interact and visualize with the data
		Downloads	Downloads of last month
		Dataset stats	Size of the dataset files, size of the auto-converted Parquet files, and number of rows
		Usage interface	Integration with several libraries to use the dataset
		Collections	Collections that contain the dataset
	Space	Demo	Demo application of a ML model
Files	All	file tree	List of files and folders
		Git data	Git commit history and branches
Community	All	Pull requests	Pull request threads
		Discussions	Discussion threads

the repository files and their commit history, while the community tab hosts the discussions and pull requests threads arisen during the development of the repository. These two tabs are the same for all repository types.

Since its creation, HFH has been rapidly evolving and incorporating new features. For instance, the discussions and pull requests tab was released in May 2022, the full-text search in February 2023,[5] collections in September 2023,[6] and Posts in December 2023,[7] similar to a social network where users can interact within publications (i.e., post) made by other users.

To illustrate the growing evolution of the platform, we relied on the Diffusion of Innovation (DoI) theory as proposed by Squire (2017), which helps to explain how a product gains or loses momentum in a system. Figures 1a and 1b illustrate the natural and cumulative growth of new project registrations by month in GitHub, which was reported by Squire (2017). For the sake of comparison, we replicate this experiment in HFH using HFCommunity (Ait et al., 2023a), shown in Figures 1c and 1d. In Figure 1d the point

[5] `https://huggingface.co/search/full-text`
[6] `https://huggingface.co/collections`
[7] `https://huggingface.co/posts`

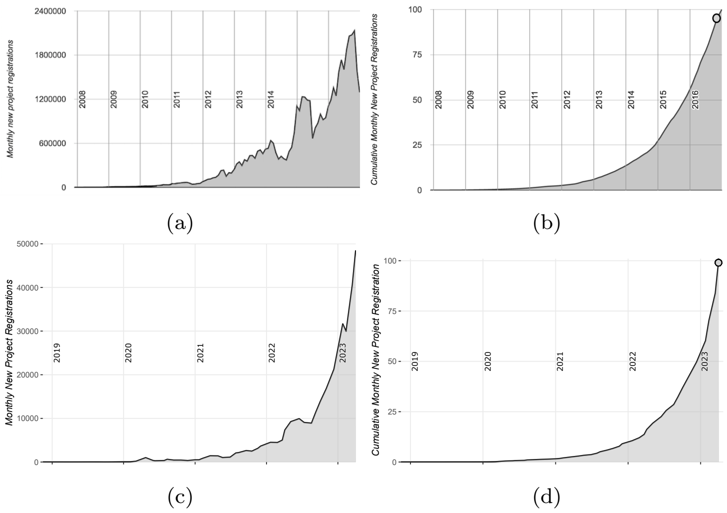


Fig. 1: (a) Monthly and (b) cumulative new project registrations in GitHub, 2008-2016 (Squire, 2017). (c) Monthly and (d) cumulative new project registrations replicated in HFH, 2018-2023.

indicates the month with the maximum growth. As can be seen, the point is located in the last month of registered activity, which indicates that no momentum lost is detected, and therefore the platform is still growing. Note that HFH follows a growth pattern similar to GitHub.

Even though HFH is showing such a growing behavior, to the best of our knowledge, the number of research papers targeting empirical studies based on the platform is still very scarce.

3 State of the Art

Many projects developed on code-hosting platforms are public, thus allowing anyone to explore their activity, which includes access to commits, issues, pull requests and comments, among others. This large amount of public data has enabled researchers to easily collect and analyze such data. As a result, many empirical studies have been conducted in the last years, in particular, mostly relying on the GitHub platform, as noticed by Demeyer et al. (2013), Cosentino et al. (2017), and Dabic et al. (2021). Nonetheless, an increasing number of empirical studies targeting HFH have started to emerge.

In particular, Kathikar et al. (2023) addresses the analysis of vulnerabilities of open-source AI, by performing a literature review to find the linkage between models hosted in HFH and its source code in GitHub. Jiang et al. (2023b) studies the reuse of pre-trained models (PTM) in HFH contributing by: (1) depicting a decision-making workflow for PTM reuse; (2) measuring the risks of collaboration in HFH and identifying potential software supply chain concerns; (3) publishing a dataset of PTM packages; and (4) identifying unique properties of PTM package reuse. Castaño et al. (2023b) examines the carbon footprint generated by HFH models. Their study investigates the reported carbon emissions of ML models during training, and how they compare in terms of carbon efficiency. They apply repository mining techniques to retrieve the set of HFH models. Castaño et al. (2023a) also studied the evolution and maintenance of HFH models. They observed the responsiveness of the HFH community to the adoption of new models, in particular in the generative AI field. Their study also provided insight in the development of ML models, analyzing their commit activity. Yang et al. (2024) leverage on the information of dataset cards to analyze community practices and norms in dataset documentation. Furthermore, they provided a set of dataset cards as a community resource. Jiang et al. (2023a) studied naming conventions in the PTM ecosystem, targeting HFH and other model hubs. They provided a taxonomy on PTM naming defects and developed a methodology and algorithm to detect them.

We also analyzed the threats to validity of the works presented (see Table 2 for a summarization) as they can have an impact on the suitability analysis we conduct in Section 6, where we further discuss these threats and how they relate to our findings. Castaño et al. (2023b) and Yang et al. (2024) noticed the **NLP predominance in HFH**. As aforementioned, HF is a company originally known for its contributions in the NLP field. However, HFH is intended to host any kind of ML artifact. In Castaño et al. (2023a) and Castaño et al. (2023b), the **rapid evolution of HFH** is considered a threat. This threat relates to the rapid growth of HFH. The continuous work and rapid adaptation to the need of the users might affect the internal structure of HFH API hampering the replication of the studies. Thus, making it mandatory to share a replication package with the data used or an external source such as HFCommunity which provides periodical data dumps. The works of Castaño et al. (2023a), Castaño et al. (2023b), and Yang et al. (2024), relied on data available in the **repository card which is populated by the users**. We want to highlight the risk of relying on data which is reported by the users themselves. Besides the easy-to-use interface provided by HFH to fill all the required metadata for a repository, some works also appeared to provide support for the definition of the metadata such as Croissant (Akhtar et al., 2024) metadata format, which has been integrated in HFH, or DescribeML (Giner-Miguelez et al., 2024) implemented as a VSCode plugin. In Castaño et al. (2023b) and Yang et al. (2024), they **rely on few attributes** to measure popularity. Further information on repositories would help characterize better the HFH repositories, such as the number of Inference API calls. Castaño et al. (2023a), Jiang et al. (2023a), and Jiang et al. (2023b) shared the threat of **relying solely on HFH**, where results found in HFH might not be generalizable to other platforms.

However, the potential perils of empirical studies on public software data are also relevant (Howison and Crowston, 2004; Kalliamvakou et al., 2014, 2016; Flint et al., 2022). Perils could involve the quality of the project's data,

Table 2: Threats to validity from presented studies involving the HFH.

Paper	Threats Identified
Castaño et al. (2023a)	Absence of standardized reporting for metadata on ML models Changes in HFH API or HFCommunity might affect the reproducibility Relying solely on HFH
Castaño et al. (2023b)	HFH NLP-predominancy Lack of information on model cards Rapid HFH evolution Relying on few attributes Relying on user-reported data
Jiang et al. (2023a)	Relying solely on HFH
Jiang et al. (2023b)	Relying solely on HFH
Kathikar et al. (2023)	Not reported
Yang et al. (2024)	HFH NLP-predominancy Relying on few attributes Relying on user-reported data

the scarce use of the platform's features or the purpose of the project, among others. This situation may affect the quality of empirical studies, but also may raise concerns about the replicability of the results (Robles, 2010). To the best of our knowledge, this type of "promises and perils" evaluation for HFH has not been yet performed. Starting a discussion on the general suitability of HFH for empirical studies is the purpose of this study.

This analysis is crucial before we start new empirical studies on top of HFH to better learn how to best develop and maintain this new breed of ML-based projects and understand their differences regarding traditional software development (Gonzalez et al., 2020).

4 Research Method

In this section we discuss how our study has been set up, which was pre-registered in Ait et al. (2023b). We first present our goal and research questions (Section 4.1), and then we report on how we plan to address each research question (Section 4.2).

To identify the goal, research questions and metrics we followed an approach similar to the Goal Question Metric (GQM) (Wohlin et al., 2012). The research questions have been formulated based on the goal of assessing the current state of HFH and analyzing its adequacy to be used in empirical studies. A detailed summary of the deviations with the pre-registration is presented in Section 4.3.

4.1 Research Questions

The goal of our study is to assess the current state of HFH and analyze its suitability to be used in empirical studies. In particular, we address the following research questions:

RQ1 What features do HFH provide as a code-hosting platform to enable and be targeted in empirical studies? We aim to comprehend the key features that characterize HFH both for individual projects (i.e., features oriented towards end-users planning to use HFH for their software development projects) and at the platform level (i.e., to facilitate the retrieval and analysis of global HFH usage information). This analysis allows characterizing the platform and identifying potential use cases for empirical studies. Thus, we subdivide RQ1 further into:

- RQ1.1 What features are typically offered by code-hosting platforms? To contextualize the analysis of the HFH features, we first need to collect the features offered by other code-hosting platforms. We study existing code-hosting platforms to define a feature framework covering their functionality at the project and at the infrastructure level. This feature framework could be used to characterize any other (future) code-hosting platform as well.
- RQ1.2 What features HFH offers to facilitate the collaborative development of ML-oriented projects? This research question performs an exploratory study of the features offered by HFH to projects hosted in the platform among those identified in the feature framework of RQ1.1. In this RQ, we focus on the features serving project development tasks, such as pull requests for managing code contributions or issue trackers for notifying bugs or requests.
- RQ1.3 What features HFH offers at the platform level to facilitate access to the hosted projects' data? In this research question, we examine the features provided by HFH aimed at retrieving its internal data, derived from the activity of its projects and their surrounding communities. Indeed, note that these features are not necessarily aimed at developing software projects in the platform (as it is the case of the features studied in RQ1.2) but at enabling the data collection from them. Furthermore, we are interested on identifying whether such infrastructure enables to collect information from each of the HFH features identified in RQ1.1. We believe the availability and easy access to the data in a code-hosting platform is a relevant factor for researchers when selecting platforms for their empirical studies.

RQ2 How is HFH currently being exploited? We are interested in studying how HFH is so far being used at platform and project levels. In each level, we analyze the data within two perspectives: volume and diversity.

To measure the volume, we define numerical variables, such as the number of repositories and users at platform level; or the number of files, contributors and commits at project level. On the other hand, to measure diversity we define categorical variables, such as the programming languages used in the repositories or the types of contributions (i.e., issues or discussions) in the projects. Note that while RQ1 focuses on the features provided by the platform, RQ2 analyzes its current usage, thus allowing to better understand the platform dynamics. We subdivide RQ2 further into:

- RQ2.1 What data-related metrics can be defined for evaluating the usage in code-hosting platforms? In this research question, we explore the metrics that can be used to describe the usage of code-hosting platforms, either at platform or project level. These metrics will be used in the following research questions.
- RQ2.2 What is the current state of the platform data in HFH? In this research question, we explore how HFH is used as a whole. Some examples of variables to be used in this research question are the number of repositories and the level of dependency between them, as an example of volume and diversity.
- RQ2.3 What is the current state of the project data in HFH? In this research question, we explore the usage of HFH at project level. Thus, instead of the platform, the repository becomes the unit of study. The goal is to characterize the average (or averages if we detect different typologies) project on HFH via the analysis of their number of files and commits, number of users, its temporal evolution, etc.

4.2 Methodology

To address our research questions, we conducted a mixed analysis of HFH. To address RQ1, we perform a qualitative analysis which focuses on identifying the features of HFH and the options available to retrieve HFH data. On the other hand, to address RQ2, we first follow a qualitative method to identify the metrics, that are then used in the quantitative analysis, which inspects the data available in HFH. Both analyses are also validated via qualitative methods (i.e., a poll and an interview), as we describe below.

Figure 2 shows the methodology we defined to address the research questions. The process consists of, first, defining a feature framework and a set of metrics, which address RQ1.1 and RQ2.1, respectively. Then, we conduct the qualitative and the quantitative analysis. The former addresses RQ1.2 and RQ1.3, while the latter addresses RQ2.2 and RQ2.3. Note that each part has its validation. On the one hand, the process of the definition of the feature framework and the set of metrics includes a survey validation, which leverages on experts (i.e., developers relying on code-hosting platforms for their

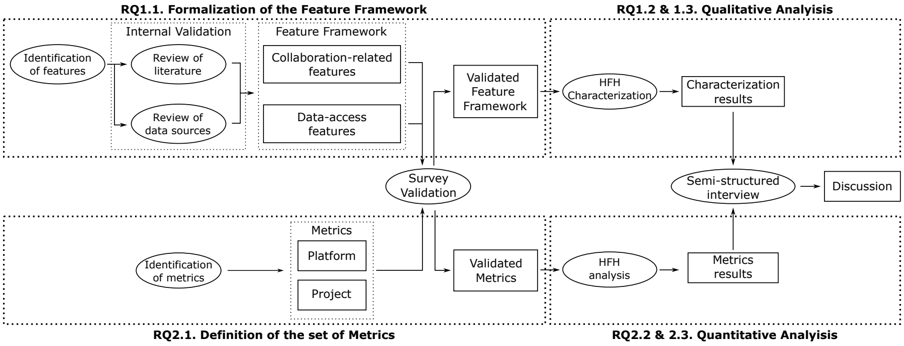


Fig. 2: Methodology to address the research questions.

day-to-day development activities and researchers of empirical research and mining software repositories (MSR) communities) to assess the adequacy of the features and metrics identified. On the other hand, the qualitative and quantitative analysis includes an interview, where participants assess the results of the analysis and prompt a discussion around those results. Next, we present our methodology in detail.

4.2.1 RQ1. HFH Characterization

We cover in this section all steps involved in the analysis of RQ1 and its subsections.

Formalization of the feature framework. To address RQ1, we first address RQ1.1 by building a feature framework aimed at identifying the characteristics which define a code-hosting platform. Features include both characteristics offered to develop software projects and functionalities aimed to retrieve data from the platform. The framework is built by analyzing different code-hosting platforms and identifying the features offered by each platform.

The first step is based on the identification of code-hosting platform features (see *Identification of features* in Fig. 2). We kick-start the framework dimensions leveraging on the author's experience, the platform usage and related work (e.g., Alamer and Alyahya (2017)). Features are organized according to topics (e.g., coding or project management), to focus on the different aspects of the platforms.

In the context of this paper we mainly focus on GitHub and GitLab due to their relevance and wide user base, but we also contemplate other alternatives in the internal validation steps (see *Internal Validation* in Fig. 2).[8] Thus, once we have the initial set of features, we review the literature to validate their relevance in current empirical studies (see *Review of literature* in Fig. 2). The review of literature is performed following four steps, namely: (1)

[8] In particular, BitBucket, Codeberg, Forgejo, GitHub, GitLab, HFH, Kallithea, Launchpad, Savannah GNU and SourceForge.

selection of digital libraries, where we select a set of well-known digital libraries with advanced search functionality and the support to export the results (e.g., as BibTex or CSV files); (2) query of papers, where we define specific queries for each feature; (3) export and process results, where we analyze and digest the results, and remove duplicate entries; and (4) report of the results.

Apart from reviewing the literature, we also cover those features aimed at retrieving platform internal data (i.e., data-access features), such as APIs or search functionalities and external data sources as community provided datasets (see *Review of data sources* in Fig. 2). The review of data sources is performed by following three steps, namely: (1) query of papers, where we define specific queries for each feature; (2) identification of data sources, where we identify the proposed solutions by the platform internal team (e.g., APIs or libraries) and the community production (e.g., datasets or tools) by querying papers of the selected digital libraries; and (3) categorization of features, where we classify the available options by its functionality (e.g., search mechanisms, real-time data access or offline data snapshots).

While the review of literature covers collaboration-related features, the review of data sources involves mostly data-access features, but both can cross-fertilize each other. These features are defined within the feature framework, resulting in a set of features to characterize code-hosting platforms. Note that while our feature framework helps characterize HFH, it may also potentially help to characterize any other current (or future) code-hosting platform. The proposed feature framework will be validated by the survey validation, as we describe below.

Survey validation. We validate the resulting set of features and metrics with experts of the field by conducting a survey with open questions (see *Survey Validation* in Fig. 2). To this aim, we created a survey divided into three sections: (1) profiling, (2) feature revision, and (3) platform and project metric revision. Sections 2 and 3 include open questions to let survey participants develop their agreement or disagreement on the features' identification. Additionally, in this survey, the participants can decide whether to later participate in the semi-structured interview to express their insights and validate the characterization and analysis of HFH.

The survey questions are defined depending on the profile of the participant, which can be developer or researcher. Regarding the developer profile, we are interested in the importance they give to the features of code-hosting platform to develop their projects. On the other hand, regarding the researcher profile, we are interested in the interest of the features to perform empirical studies. Therefore, we can extract a combination of insights of: (1) participant's preference to the features when selecting code-hosting platforms (to develop projects or to perform empirical studies), and (2) which features are more present in the participants' working day, as it may identify relevant features, or it may have more use in the code-hosting platforms.

The agreement on the features is reported with a Likert scale (from 1 to 5), where participants report whether they think the feature is more appropriate or not. Each group of features (e.g., coding or platform management) has an open question in order to let authors give their insights on the feature identification such as wrong categorization, lack of specific concepts or proposal of new features.

The participants of the survey are practitioners of the empirical research and MSR communities and/or active developers relying on code-hosting platforms, which will be either contacted by e-mail or social networks. With the agreement and the insights provided by the participants, we started building the validated feature framework.

HFH characterization with a qualitative analysis. From the *Survey Validation* we obtain the *Validated Feature Framework*, which is the curated version of the structure of the previous set of identified features. This validated feature framework is intended to be used as a reference framework to analyze the HFH platform. Thus, we characterize HFH using the framework (see *HFH characterization* in Figure 2), which will result in two sets of features: (1) absent features or (2) available features. The interpretation of the results is comprised by the following steps: (1) coverage/range of features, we identify whether HFH lacks or not in some aspects (e.g., coding support or project management features); and (2) reach of features, from those features that HFH provides, to which extent it covers its topic (e.g., replies, reactions and mentions in forum-like threads). This characterization process is discussed with a semi-structured interview (see *Semi-structured interview* in Figure 2).

4.2.2 RQ2. On the usage of HFH

We now cover all steps involved in the analysis of RQ2 and its subsections.

Definition of the set of metrics. To address RQ2.1, we need to examine the HFH data to provide an overview of the current usage of the platform. We plan to study the usage of HFH at platform and project level. The former shows the actual usage of the features identified in the previous research question and conclude on the level of exploitation of such features. The latter provides an insight of how the development process is currently carried out in HFH, thus favoring the comprehension of why the users use this platform.

To perform the analysis, we first define a set of metrics to analyze the data (see *Identification of metrics* in Fig. 2). The selection of metrics is based on the authors' experience in reading and performing a significant number of empirical studies, and meta-studies (Cosentino et al., 2017), paired with their experience in using also the HFH.

For the sake of clarity, we have selected the metrics we believe are most interesting for evaluating the size and diversity of data behind a code hosting platform (HFH in particular) but, of course, more metrics could be added.

While metrics aim to be generic, given our special interest in HFH, we also specialized some to target HFH specificities. For instance, in HFH, we can easily study the nature of a repository type (i.e., a pre-trained model or a dataset) or the dependencies between the repositories of different types, such as how many datasets are used by multiple models.

Survey validation. As aforementioned, the survey validation step is used to validate both the features and the metrics. For RQ2, the focus of the validation is on the metrics. Therefore, we surveyed the identified metrics to understand which metrics contribute to the selection of a platform by both developers and researchers. While developers may pay more attention to metrics that are more related to the platform usage (e.g., number of users in the platform), researchers may focus on metrics where open source development is more characterized (e.g., conversations in issue's threads). This dual view allows us to better understand and define a set of metrics as broad as possible.

HFH quantitative analysis. After the survey validation, we confirmed the set of *Validated Metrics* that will then be used for the analysis of HFH. The analysis is split into the two types of metrics: (1) platform-level analysis and (2) project-level analysis. While platform-level analysis is proposed as way of studying the nature of HFH environment, project-level analysis allows giving an insight on the status of the repositories hosted in HFH.

Both analyses (see *HFH analysis* in Figure 2) are conducted according to the following steps: (1) selection of a data source where we choose the most suitable option to retrieve HFH data; (2) extraction of HFH data, this step prepares the required software to mine HFH data; (3) curation of data, in order to exploit the data we have to apply, if necessary, some data curation techniques (e.g., format conversion or data cleaning); (4) data description, where we report the metadata of the curated data; (5) metric calculation, this step queries the data to extract the validated metrics; and (6) metric visualization and interpretation, in order to examine the calculated metrics.

The metrics enable us to analyze the usage of HFH, both at the platform level and at the project level. Following common practices when reporting quantitative data, we report the average and the standard deviation, except when the data is skewed. As there is little consensus when reporting descriptive statistics for very skewed data, we report median, Inter-quartile Range (IQR), average, and standard deviation for the sake of clarity. This usage analysis is also discussed with a semi-structured interview (see *Semi-structured interview* in Figure 2).

4.2.3 Semi-structured Interview

The conclusions we extract from both research questions, along with a broader view of HFH, are discussed in a semi-structured interview, allowing interviewees to explore particular themes or responses further (Wohlin et al., 2012). Our objective is to guide the interviewee with a set of questions, proposing specific questions for either researches or developers. Distinguishing between these two groups favors the robustness of the validation, as researchers might have more knowledge on methodologies techniques and research plans, and developers might have valuable experience in industrial situations, thus improving the chance of possessing a more pragmatic vision of the results (i.e., characterization and analysis). The results of the interview prompt discussions, which we review in Section 6.

Following the suggestion of Wohlin et al. (2012), the interview is organized into four steps, namely: (1) presentation of objectives, where we present the objective of the interview, how the data from the interview will be used and a briefing of our conclusions of the paper, along with some guidance; (2) introductory questions, where we ask about our conclusions of the paper and seek opinion on the agreement or disagreement; (3) general questions, where we ask about their perception on the role of HFH, seeking discussion on the different points of view; and (4) report of results, where we interpret the interview and synthesize discussion points.

The first step is provided prior to the interview via e-mail. The proposed questions of second and third steps are presented in Table 3. The table lists the code and specific question of the interview. The question code format is composed by the question type, the question number, and the profile of the interviewee. Question type refers to either introductory questions ("CH" for RQ1, regarding HFH characterization; and "AN" for RQ2, regarding HFH usage analysis) and general questions ("GQ"); while the profile is indicated by an "R" for researchers and "D" for developers.

The interview will be conducted by two interviewers, as it might indicate more communication by the interviewee (Hove and Anda, 2005), encouraging interviewees to elaborate more on their answers. All interview subjects received information about the purpose of the interview, along with the topic of our research, the expected duration, the disclaimer and agreement about the recording of the interview, the format of the semi-structured interview, and a draft of our paper results to analyze our conclusions. Interviews are online and last 30 minutes. Interviewees are volunteers from the survey and external candidates, contacted by e-mail. They are described in Section 5.5.

The result of the interview, besides reviewing our process, leads to a discussion about the suitability of HFH as a source for empirical studies (see Section 5.5 and Section 6, respectively).

4.3 Summary of Deviations

While addressing the research questions and executing the study we noticed some circumstances that required a deviation from the pre-registered research plan. The following list enumerates the deviations.

- We extended both RQs adding a new sub-question (i.e., RQ1.1 and RQ2.1). These two new sub-questions address the need for a reference framework to answer the original sub-questions (i.e., RQ1.2 and RQ1.3, and RQ2.2 and RQ2.3).

Table 3: Reference questions used by the interviewer.

Code	Question
CH1	Do you agree with our characterization of HFH? Do you differ in any conclusions?
CH2	Were you surprised when seeing our conclusions? Does the idea of HFH prior to seeing our conclusions remained the same?
CH3R	Does HFH provide enough features to perform your empirical studies? Do you miss any feature? Which feature, or features, do you have analyzed?
CH3D	Does HFH provide enough features to select a suitable model or dataset? Do you miss any feature? Which feature, or features, do you rely on more when developing in code-hosting platforms?
CH4	What do you think about the data access mechanisms? Have you ever used any? Would you like to have an alternative option?
AN1	Did our analysis help you to understand the current state of HFH? Which metric surprised you the most? Do you disagree with any metric interpretation? As concluded in RQ2, do you feel that the usage of HFH is well reported? Does it reflect the reality of HFH?
AN2	What other analysis of HFH data would you like to see?
AN3R	Do you think previous studies on GitHub, or other platforms, could be applied in HFH? Do you think any previous study that you performed could be replicated with HFH data?
GQ1	What do you think about the future of HFH, will it become a replacement source of GitHub or will it be used as a complementary source?
GQ2	For your next new project, would you use GitHub, HFH, or another platform? What do you look most when selecting a platform?
GQ3	What do you think HFH needs to do to attract you, or others, more to the platform?
GQ4	How do you see the community/open-source development of AI artifacts in 5 years?
GQ5	Would you like to add anything to the discussion?

– We introduced a second validation step. The first validation step (i.e., *Survey validation*) addresses the validation of the artifacts we defined to analyze HFH (i.e., the feature framework and metrics) via a survey, while the *Interview validation* step conducts a semi-structured interview to discuss the findings of the RQs and prompt discussion, while also trying to mitigate the threats to validity from the creation of the artifacts (see Section 7.2).

5 Results

In this section we present the results of our study. For each research question, we refer to specific steps of Figure 2 when presenting the results.

5.1 RQ1.1. Formalization of the Feature Framework

Identification of features. This step resulted in a list of 31 features divided into six categories, which have been identified by authors' knowledge of the platforms and leveraging on similar works, as stated in Section 4.2.1. The categories are: coding, social, user management, project management, project add-ons and data access.[9] Table 6 visualizes the categories and metrics identified. Note that this table shows the already validated feature framework. Therefore, to clarify better the evolution of the selected features, we identified in gray and italicized the discarded features (i.e. features that were not validated), and we provide the agreement column in order to understand the results of the *Survey validation*. We provide more details about the validation in the *Validated Feature Framework* step.

Review of literature. We selected three digital libraries, namely: (1) IEEE Xplore,[10] (2) ACM DL,[11] and (3) ScienceDirect.[12] These libraries offer advanced search functionality, ability to export the results to a common format (i.e., BibTex), and its relevance in the research field. We queried these digital libraries to collect papers written in English and with titles including the platform name.[13] We found a total of 621 references. Interestingly enough, we only found results for four platforms: GitHub, GitLab, HFH and SourceForge, thus discarding the rest for the remainder of the review of the literature. Table 4 provides the number of results of each platform. Note that the number of results for GitHub is significantly higher than for the rest of platforms. To cope with this, we collected articles written in English and having both the feature and the platform name in its title.[14] Table 5 provides the number of results of each feature in GitHub. We found 124 unique papers, with an arithmetic mean of articles per feature $\mu = 6.53$ ($\sigma = 11.38$).

Given the low number of papers collected for platforms different than GitHub, we performed a manual evaluation to identify empirical studies targeting any specific feature. From all 12 papers retrieved from GitLab, three are empirical studies addressing features covered in our framework (e.g., targeting CI/CD workflows (Fairbanks et al., 2023), projects and users networks (Safari et al., 2020) or stream analytics (Eraslan et al., 2020)). The remaining papers are either tool proposals or methodologies for CI/CD, or education related topics (e.g., plagiarism). Papers targeting HFH do not address any specific feature, but rather the platform as a whole (e.g., the work by Jiang et al. (2023b) presents an empirical analysis on the PTM reuse in the platform). Finally, papers analyzing SourceForge mainly report on empirical studies to enrich information on software repositories (e.g., software descriptions (Bäumer et al., 2017), reflexivity (Foushee et al., 2013) or success of projects (English and Schweik, 2007)).

Regarding the papers analyzing GitHub and specifying the feature name in the title, we see a high presence of papers addressing features such as issues

[9] This study was performed on October, 2023.
[10] `https://ieeexplore.ieee.org/Xplore/home.jsp`
[11] `https://dl.acm.org/`
[12] `https://www.sciencedirect.com/`
[13] Queries available in Appendix 8.1.1
[14] Queries available in Appendix 8.1.2

Table 4: Number of papers collected from each digital library including the platform name in the title.

Platform	ACM	IEEE	SciDir
GitHub	253	299	43
GitLab	6	5	1
HuggingFace Hub	1	3	-
SourceForge	5	5	-

Table 5: Number of papers collected from the digital libraries including GitHub and the feature name in the title.

Feature	# Papers	Feature	# Papers
Issues	53	Pull Request	14
Development Workflow	7	Fork	7
Release	7	Code Review	5
Licensing	4	Snippets	4
Tagging	4	Branches	3
Following	3	Proj. Relations	3
Packages	2	Roles	2
Q&A	2	Collab./Cloud Cod.	1
Groups	1	Marketplace	1
Web Publish	1	CVS	0
External Integrations	0	Milestone	0
Repo Type	0	Security	0
Stream Analytics	0	Webhooks	0
Wiki	0	Work Management	0

and pull requests. Also, we find relevance in features such as development workflows (e.g., CI/CD), forking and release, all of them with seven articles; code review with five articles; and tagging, snippets and licensing with four articles. On the other hand, we detected a lack of literature references for nine features (see the last rows of Table 5), which are candidates of features to be removed. However, the removal of these features is postponed until we collect the insights of the survey validation step.

Review of data sources. To identify data-access features, we looked into different types of data sources for the code-hosting platforms, either provided by themselves or as external curated resources. To do so, we first query the three digital libraries selected in the review of literature.[15] We encountered results of dataset and tool papers. Regarding dataset papers, we found eight papers, all targeting GitHub, of either curated dataset samples for empirical studies (Yu et al., 2018; Joshi and Chimalakonda, 2019; Spinellis et al., 2020) and for training ML models (Golzadeh et al., 2021), topic-specific datasets (such as UML models from GitHub (Robles et al., 2017)), a dataset generator (Özçevik and Altay, 2023) and a meta-analysis of the methodology, data sources and limitations of selected research papers (Cosentino et al., 2016). This last article identifies several data sources from 93 research papers, where

[15] Query available in Appendix 8.1.3

they identified three main topics that may help to identify features, namely: (1) GitHub API, (2) GitHub Search API, and (3) curated datasets (i.e., GHTorrent (Gousios and Spinellis, 2012), GitHub Archive,[16] and BOA (Dyer et al., 2015)).

With regard to tool papers, we found 17 papers. As this query was targeting a keyword with a broader meaning, we only report the papers presenting tools to mine and extract data from the platforms.We found seven papers which describe tools to (1) mine the platform's data (Romano et al., 2021; Pina et al., 2022; Valenzuela-Toledo et al., 2023), (2) to process commit's data (Casalnuovo et al., 2017), (3) to visualize data (Kaide and Tamada, 2022), and (4) to extract data from HFH (Ait et al., 2023a) and from SourceForge (Kritikos and Chatziasimidis, 2011).

From all analyzed options, we only found HFCommunity (Ait et al., 2023a) as a data source specifically targeting HFH. HFCommunity is an open-source tool that collects data from HFH and Git repositories, and stores it in a relational database to facilitate their analysis. In Section 3, we described the work of Jiang et al. (2023b), which publishes their dataset named HFTorrent that contains a snapshot of 15,913 PTM packages from five model hubs. In contrast to this proposal, HFCommunity provides data from all public repositories hosted in HFH.

The analysis of these works allowed us to extend the features characterizing data sources. We identified four features (i.e., search, API, integrated CLI and datasets) which will define the data access topic. Note that *datasets* feature should not be confused with dataset repositories of HFH.

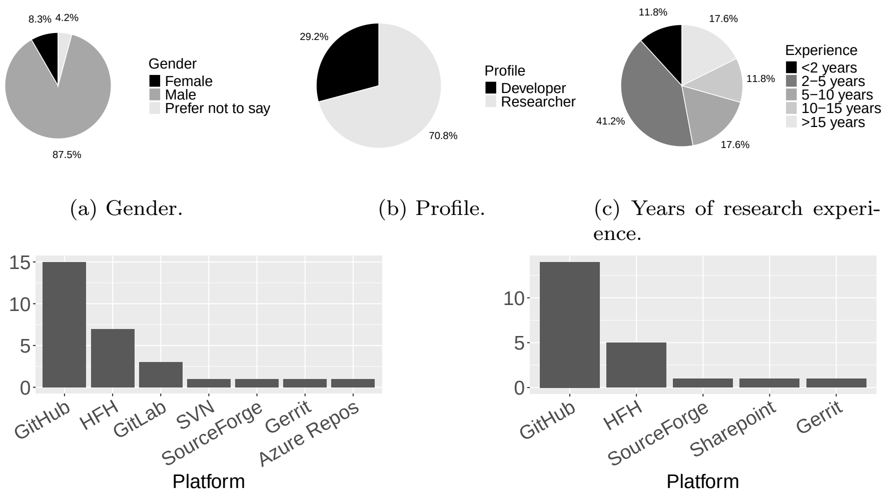
These two review processes are turned into the proposed feature framework for the *Survey Validation.* This framework is composed of 32 features categorized into six categories.

Survey Validation. 24 participants answered the survey. Figure 3 provides basic profile information of participants. The majority of participants are males and work in research. Regarding researchers, most of the participants have more than ten years of experience (see Figure 3c), and report the use of GitHub to develop their work, both for developing and performing empirical studies (see Figure 3d and 3e, respectively). Note that developers can only report the platform they use in development.

As described in Section 4.2.1, we asked participants to rate (from one to five) the relevance of the features in the framework. Thus, we induce a relevance indicator from the median of the results, as commonly used when aggregating Likert results (Joshi et al., 2015). Those features with no literature and with a relevance below three, are discarded (see Relevance column in Table 6). All features have a relevance indicator of three or above. Thus, we do not discard any feature at this point.

We also evaluate the comments provided by the participants, which provided evidences to consider renaming and moving features across topics. In

[16] `https://www.gharchive.org/`



(a) Gender. (b) Profile. (c) Years of research experience.

(d) Platforms used in development. (e) Platforms targeted in empirical studies.

Fig. 3: Participants profile.

particular, we moved three features and renamed one feature: (1) *Website publishing* was moved from *Coding* to *Project Management* topic, as stated from a participant: "*... I wonder if "Website publishing" should be actually categorized as "Coding feature". It looks more like a deployment-related feature than a code-related feature.*"; (2) *Security* was moved from *Project Management* to *User Management* topic, as stated from different participants: "*Should security be under users?*" and "*... My understanding is that security is an attribute of the software project itself rather than its development life cycle.*"; (3) *Issues* from *Social* to *Coding* topic, as we believe they are more used in coding related tasks, also stated by a participant: "*I don't know if "Issues" should be categorized as a "social feature". I acknowledge that issues are meant for discussion, however.*"; and (4) *Security*, was renamed to *Permissions* as some participants indicated: "*Granular control of user permissions and group / teams ...*". We performed these changes to provide a well-structured feature framework which can potentially be used to analyze any code-hosting platform.

Comments from participants also helped us to identify new features. In particular, we added two features: (1) *Code navigation*, in the *Coding* topic as stated from a participant: "*Browsing/searching through* GitHub *repositories ... has made less arduous the task of reading code. Also, 'go to definition' has been incorporated*"; and (2) *Social profile* in the *Social* topic as stated from a participant: "*... "Collaborators" feature where you can manage the project contributors and, also, see their user profiles (job, title, etc.) ...*".

We apply the *"Review of literature"* step to the three new features, resulting in *Social profile* having five papers where it is targeted GitHub's profiles (e.g.,

Table 6: Validated feature framework (removed features in gray and italicized).

Topic	Feature	Description	Relevance
Coding	CVS	Control version system (e.g., Git, Mercurial, Subversion, etc.)	4
	Forking	Creation of a copy of other projects	4
	Issues	Reporting of bugs and requests	5
	Pull Request	Submission of contributions to other projects	5
	Code Review	Discussion about changes in project files	5
	Release	Identification and management of deployable software iterations	4
	Packages	Management and release of GitHub packages	4
	Snippets	Upload of fragments of code to share (e.g., GitHub Gist)	3
	Branches	Navigation and management of CVS branches	5
	Collaborative/ Cloud coding	Online development of project files (e.g., GitHub Codespaces, HTH resources)	4
	Code navigation	In-file search utilities such as linking definitions and entity's reference	New
Social	Q&A	Discussions	4
	Following	Support for stars and following platform users	4
	Social profile	Support for personal information of users (e.g., job, title, etc.)	New
User Mgmt.	Groups	Support for defining teams of users in projects	4.5
	Roles	Roles inside the repository	5
	Permissions	Support for granular control of user permissions and groups	New
Project Mgmt.	Milestone	Similar to Coding / Release	3.5
	Wiki	Wiki-based system for project's documentation	3
	Work management	Agile-like boards to organize tasks (e.g., GitHub projects; GitLab To-do lists)	4
	Stream analytics	Project insights (e.g., GitHub analytics and repository insights)	3
	Tagging	Project's tag definition and management	3.5
	Security	*Access control to project's assets (e.g., visibility, code control, etc.)*	4
	Licensing	License identification for projects	4
	Development workflows	Continuous Integration and Development	4
	Project relations	Definition of link between projects (e.g., dependencies, etc.)	3
	Repository type	Classification of projects according to their purpose	3.5
	Website publishing	Support for hosting websites (e.g., GitHub Pages)	3.5
Project Add-ons	Webhooks	Integration with external applications (e.g., GitHub Actions)	5
	External integrations	Support for integration with external services (e.g., Campfire, Jira, Slack, or social networks)	4
	Marketplace	Catalogue of external integrations	4
Data Access	Search	Search function for platform assets (e.g., repositories, files, users, etc.)	N.A.
	API	Support for accessing the platform programmatically	N.A.
	Integrated CLI	Tool to interact with the platform from the command line	N.A.
	Datasets	Existing datasets to query the platform	N.A.

Hauff and Gousios (2015) and Gajanayake et al. (2020)) while *Code navigation* and *Permissions* do not have any result.

Finally, we updated the definition of *External integrations* feature to be more understandable as it caused some confusion among participants: "*... integration with external social networks (but maybe this feature is addressed in the Project add-ons features section)*"

The resulting feature framework is composed by 34 features categorized into six categories.

Validated Feature Framework. Table 6 shows the validated version of the feature framework, used to perform the characterization (i.e., qualitative analysis) of HFH. The first five topics, namely: coding, social, user management, project management and project add-ons address RQ1.2; while the last topic (i.e., data access) addresses RQ1.3. The last column corresponds to the relevance indicator of developers and researchers from the survey validation step. Features removed are shown in gray and italicized while new features are identified as *New* in the relevance column. In the following, we describe the topics, motivate them in the context of empirical studies and report relevant literature targeting their features.

Coding. This topic includes variables addressing typical developers' needs to perform coding tasks, namely: usage of a version control software, and support for forks, issue trackers, pull requests and code review, where they facilitate user communication during development. These features conceive the pull-based development model (Gousios et al., 2014) and has allowed the execution of empirical analysis on forking (Biazzini and Baudry, 2014; Ren et al., 2018), issues (Destefanis et al., 2018; Wu et al., 2022), pull requests (Yu et al., 2015; Chen et al., 2019) and code reviewing (Wessel et al., 2020; Al-Rubaye and Sukthankar, 2023). Additional features facilitate the development of projects (i.e., releases, package encapsulation and upload of code snippets), the navigation through CVS branches and code, and online development (i.e., collaborative and cloud coding). These features are also targeted in empirical studies on software releases (Eibl and Thurnay, 2023), software packages (Decan et al., 2016), code snippets (Baltes et al., 2017), branching (Zou et al., 2019) and cloud coding (Malan, 2022).

Social. This topic includes variables identifying user interaction such as the creation of Q&A threads, the ability to follow and like projects and the profiling of users. Addressing this kind of features has enabled studies on how users participate in discussions (Tsay et al., 2014) or the importance of social indicators (i.e., stars or follows) in code-hosting platforms (Borges and Tulio Valente, 2018).

User management. This topic is related to the ability of creating and managing groups of users with the purpose of sharing projects between multiple users. Inside groups, a hierarchy structure can appear, thus defining roles inside the groups of users or inside a specific repository. Furthermore, we acknowledge the support for management of the users' permission as a security concern. These features are targeted in studies on the impact of large organizations (i.e., groups) (Lazarine et al., 2022) and the technical roles of GitHub users (Montandon et al., 2021).

Project management. The projects, or repositories, are the root element in social code-hosting platforms. This topic includes the study of support for management tools such as milestones, agile-like boards, stream analytics, tagging or labelling, and website publishing services, and the support for development workflows (e.g., CI/CD). It also includes project identification features such as licensing, project dependencies, and categorization of repositories depending on their purpose (e.g., documentation repositories). These kinds of features have been used in empirical studies addressing tag assignment of repositories (Cai et al., 2016), issue labelling (Wang et al., 2022), continuous integration in GitHub projects (Baltes et al., 2018), licensing inconsistencies in GitHub (Wolter et al., 2023) and project dependencies updates (He et al., 2023).

Project add-ons. Social code-hosting platforms usually allow projects to integrate with apps, available via a marketplace; and communicate with external services via webhooks (e.g., GitHub Actions). This topic covers these features, which have enabled empirical studies on the GitHub marketplace (Souza et al., 2021).

Data access. This topic covers those auxiliary and technical tools to enable the collection of data for empirical studies. Thus, they aim at facilitating the use of the platform, including the existence of a platform API, an integrated CLI to access the platform or the indexation of the platform content, such a search mechanism. Furthermore, we consider the existence of external datasets gathering data from the platform.

5.2 RQ1.2 & 1.3. HFH Characterization

Once defined the feature framework, we proceed with the *HFH characterization*. We first aim to identify the features of our framework within HFH. This results into two sets of features, namely: absent features and available features. We detected 15 absent features and 17 available features. Table 7 shows the characterization of HFH according to the feature framework. Absent features are noted with a cross mark.

Next, we interpret the results by the coverage and range of the HFH features, thus revealing to which extent they cover its topic.

Coding. The coding support is limited, reduced to the minimal tools required to develop in a collaborative way (i.e., CVS, branches, and pull requests). For instance, HFH does not provide a feature for forking repositories from HFH but instead provides some workarounds such as relying on Git LFS pointers[17] or using a space to duplicate a repository without the Git history.[18] Furthermore, the pull request and collaborative/cloud coding features are a simplified version of the GitHub and GitLab propositions. Regarding pull requests, they leverage on Git References,[19] but there is in-platform support for creating them with easiness. The collaborative/cloud coding feature is provided as online file editing and support for performing inferences in models in the browser. Coding features might not be their priority, as HFH's main focus is to "explore, experiment, collaborate, and build ML technology",[20] and not code. We believe that their focus on ML artifacts redefine the development process

[17] https://huggingface.co/docs/hub/en/repositories-next-steps#how-to-duplicate-or-fork-a-repo-including-lfs-pointers
[18] https://huggingface.co/spaces/huggingface-projects/repo_duplicator
[19] `https://git-scm.com/book/en/v2/Git-Internals-Git-References`
[20] `https://huggingface.co/docs/hub/index`

Table 7: Characterization of HFH.

Topic	Feature	Comment
Coding	CVS	Git is used as a control version system
	Forking	×
	Issues	×
	Pull Request	Simplified version (i.e., fork not required)
	Code Review	×
	Release	×
	Packages	×
	Snippets	×
	Branches	Dropdown in the files and versions tab
	Collaborative/ Cloud coding	Files can be edited online, e.g., via API, to deploy and perform model inferences. It is not a cloud-based development environment (e.g., GitHub Codespaces)
	Code navigation	×
Social	Q&A	Named discussions in the community tab, along with the pull request
	Following	Repositories can be liked and users can be followed
	Social profile	Website, interests, blog posts (either from official HF or community blogs), link to papers authored (e.g., from ArXiv) and a brief introduction for profiles
User Mgmt.	Groups	Named organizations for companies, universities and non-profit organizations
	Roles	×
	Permissions	×
Project Mgmt.	Milestone	×
	Wiki	×
	Work management	×
	Stream analytics	×
	Tagging	Tags for repositories (e.g., ML task, languages targeted, libraries used or dependencies)
	Licensing	Define and display license of models and datasets
	Development workflows	×
	Project relations	Links between models, datasets and spaces can be defined
	Repository type	Three types of repositories (i.e., models, datasets and spaces), each with its own definition, presentation and features
	Website publishing	×
Project Add-ons	Webhooks	Available since February 2023
	External integrations	×
	Marketplace	×
Data Access	Search	Full-text search utility with filtering across models, datasets and spaces
	API	Inference API to use trained models to make predictions, and Hub API to interact with the platform
	Integrated CLI	Provided as part of their Python library
	Datasets	Few available datasets to query and perform empirical studies on the platform (cf. Section 5.1)

identified on other platforms (e.g., pull-based development), being other aspects more important such as the inferences, the user's interactions, and the fine-tuning of models.

Social. Indeed, they provide all features related to the social topic, allowing the interaction among the platform's users. Discussions are a response to the need of the interaction within the community, used by contributing with feedback, opinions, or bugs. This feature is key due to the nature of code-hosting platforms, where the communication is crucial to develop software in a collaborative way. Furthermore, following and social profile features,[21] bring support to track repositories, organizations and users, which might potentially create social networks.

User management. User management is not fully supported, as HFH only brings complete support to the creation of groups and named organizations. Organizations are a group of platform users, named members, which can be managed to have different responsibilities, identified with their roles. This allows companies, universities, research groups and other kind of organizations, to manage their team inside the HFH platform. However, the roles and permissions can only be set inside organizations. At the repository level, there is no support to define roles nor permissions to specific users. As an alternative, repositories can be set to private, which restricts the access to the public, or to gated, which requires the users to introduce their personal information in order to access the repository.

Project management. This topic is also slightly covered, as it only brings support for tagging, licensing and repository type and relations. Tagging is deeply covered, providing a suitable format of identifying repositories by its purpose (e.g., ML task), the libraries used (e.g., pytorch or transformers), the datasets where it retrieves the training data (i.e., a dataset repository of HFH or an outside source), the natural languages targeted (e.g., English or Chinese), the licenses used, or other tags such as carbon emissions or inference endpoints. Due to the large number of models and datasets, and the variety of those, tagging brings valuable information in the categorization of repositories in HFH. License support is also displayed in the tagging feature, as it has a section of tags specifically designed for licenses. They are also displayed in the repository's card, similarly as in other platforms. One of the strongest points of HFH is the repositories' specialization on three types (i.e., model, dataset and space), providing specific support for each type (see Section 2). Each kind of repository can also depend on others, such as a model trained with a dataset available in HFH, or a space showcasing a model repository.

Project add-ons. While there is no integration with external applications nor social networks, the support for webhooks is provided as an alternative. Hence, the integration relies on the end users. Although HFH provides built-in support for two SDKs to build spaces (i.e., Streamlit and Gradio), there is no integration with external services in models and datasets, thus being a point where HFH can improve.

[21] An example of a complete social profile: `https://huggingface.co/clefourrier`

Data access. HFH provides a fair range of features to retrieve and interact with platform's data. The search functionality allows filtering options and a new full-text search that not only finds repositories by its name, but it also examines the repository's files. The coverage of API endpoints is also quite complete, the Hub API (i.e., REST API to interact with the platform) is under continuous development, incorporating new endpoints and features. There is also a Python client library, named `huggingface_hub`, and the `huggingface-cli` to facilitate the access to the API endpoint via Python or terminal, respectively. Hence, the users can download and upload files from and to HFH, manage their repositories, run inference on deployed models, search and retrieve repositories' data, among others. Furthermore, there are internal and community produced datasets gathering data and metrics of repositories hosted in the HFH.

Answer to RQ1.1: Table 6 provides the feature framework, composed of 34 features classified into six categories.
Answer to RQ1.2: HFH provides limited support to coding, user and project management features, as it is mainly focused on developing and sharing ML artifacts. The platform mainly promotes social support, with emphasis on discussions, while it has a short range of project add-ons.
Answer to RQ1.3: All features from the data access dimension are covered in HFH. They provide full-text search, two APIs (i.e., Hub API and Inference API), and a Python client library. Furthermore, there exist a few datasets from previous empirical studies (e.g., PTM dataset from Jiang et al. (2023b)). However, HFCommunity is the only solution to provide all accessible data from HFH.

5.3 RQ2.1. Definition of the set of metrics

Identification of metrics. To address RQ2, we define 12 metrics, covering platform (i.e., four metrics) and project (i.e., eight metrics) dimensions. Table 8 shows the metrics, their type (i.e., either quantitative or categorical), a brief description definition and the relevance reported during the survey validation, which we will comment below. The metrics of the category *Platform* address the RQ2.2, while the metrics of the category *Project* address RQ2.3. Because of the nature of HFH, the proposed metrics target some specific concepts of the HFH platform (e.g., the presence of different types of repositories) along with broader concepts such as the size of the community behind a project. These metrics are proposed for validation in the next step.

Survey Validation. We evaluate our set of metrics with the survey participants to know their insights on which indicators they believe help boosting the platform visibility (i.e., platform metrics) and what they look for when starting to contribute on a project (i.e., project metrics). The last column of Table 8 shows the relevance value of the metrics. As we did in the feature

Table 8: Validated metrics used in RQ2 (removed metrics in gray and italicized).

Category	Variable	Type	Description	Relevance
Platform	Number of repositories	Q	Amount of projects in the platform	4
	Diversity of repositories	C	Project distribution by category	4
	Number of users	Q	Amount of users in the platform	4.5
	Dependency of repositories	C	Communities in the dependency graph	3
Project	Activity	Q	Activity events over time	4
	Age	Q	Time span of project's life	3
	Content	C	Distribution of repository file composition	4
	Involvement	Q	Amount of contributors	4
	Interactions	Q	Communication events	3.5
	Artifact type	C	ML task addressed by the proposed artifact	4
	Dependent Repositories	*Q/C*	*Amount and type of dependent repositories*	*2.5*
	Popularity	Q	Amount of likes and downloads	4

Q: Quantitative. C: Categorical.

framework, those metrics with an agreement below three will be discarded. Only *Dependent Repositories* is discarded.

The most relevant metric for the platform is the number of users with a relevance of 4.5. On the other hand, there is no metric standing out for projects. However, we believe the metrics with a relevance of 4 align with traditional approaches to measure success.

We also asked about what more insight of HFH data they would like to see. Although responses did not propose new metrics, some participants shared their concerns on some specific metrics, in particular: (1) evolution and usage in time, as in "*how often the artifacts ... change over time*", "*history track of ML models, version and datasets*", "*Their evolution and usage as newer models get released*" or "*personal portfolio evolution*"; (2) interdependency of repositories, as in "*tree or graph of models...to track the base model that derived to the subsequent ones*" or "*... the interdependencies of artifacts and how this could impact the maintenance and evolution of them over time*"; and (3) repository statistics, as in "*... more performance metrics, user comments and reviews, industry and academic collaborations, and the release history*", "*... task-specific metrics/benchmarks results for all models ...*", "*how diverse is the group of people who change these artifacts ...*", among others.

5.4 RQ2.2 & 2.3. HFH Usage Analysis

Once validated the metrics, we begin with the *HFH analysis*. We first select a data source. For this, we selected HFCommunity as (1) it provides both HFH and Git tracking history data; (2) its data can be downloaded as a SQL dump which enables offline querying; and (3) it provides a conceptual schema which helps on defining the queries.[22] We then report the results of the metrics for the

[22] `https://som-research.github.io/HFCommunity/diagram.html`

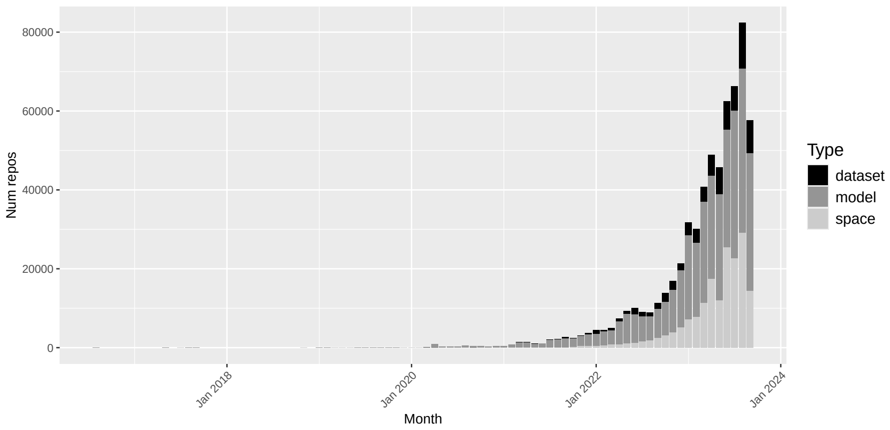


Fig. 4: Number of repositories created each month by type.

platform category, addressing RQ2.2. Regarding the *number and diversity of repositories* metrics, as of October 2023, there are 636,072 public repositories, being 381,240 models; 78,359 datasets; and 176,473 spaces. Figure 4 shows the number of total and of each type of repository over time. We see an exponential curve, being models the ones with more predominance. Given the high variety of repositories, we build three reference groups of repositories to report our results: all repositories, top-100 downloaded and top-100 liked. Repositories included in top-downloaded and top-liked are listed in Appendix 8.2.

With regard to *number of users* metric, there are 234,422 users in the platform, where 191,850 own at least one repository.

Finally, the analysis of *dependencies of repositories* metric shows that 33,872 repositories depend on another one hosted in HFH (i.e., 5.33% of the total number of repositories in HFH). Dependencies usually appear on models using a dataset (94.69% of the dependencies), but also on datasets complementing other datasets (4.30%), and spaces depending on models or datasets (0.84% and 0.34%, respectively). However, it is important to note that not all repositories report the link in the card data. For instance, spaces are typically created to showcase models, and it is rare they showcase models not also hosted in the HFH. Note also that sometimes the API is not returning a dependency even when it is explicitly stated in the repository page.

The most referenced repositories are `glue`, `squad`, and `mozilla-foundation/common_voice_7_0`, all of which are datasets, with 2,027, 1,609, and 1,323 dependent repositories, respectively. To visualize the clusters of dependent repositories, Figure 5 shows a dependency graph including those repositories with one or more dependency links. Graph nodes represent repositories, and their size is proportional to the number of dependent repositories. We show the name of the main repository in the cluster for the larger ones.

Regarding the top-100 repositories, 12 of the top-liked depend on others, all being models using a dataset, but only 16 of them are used by other projects,

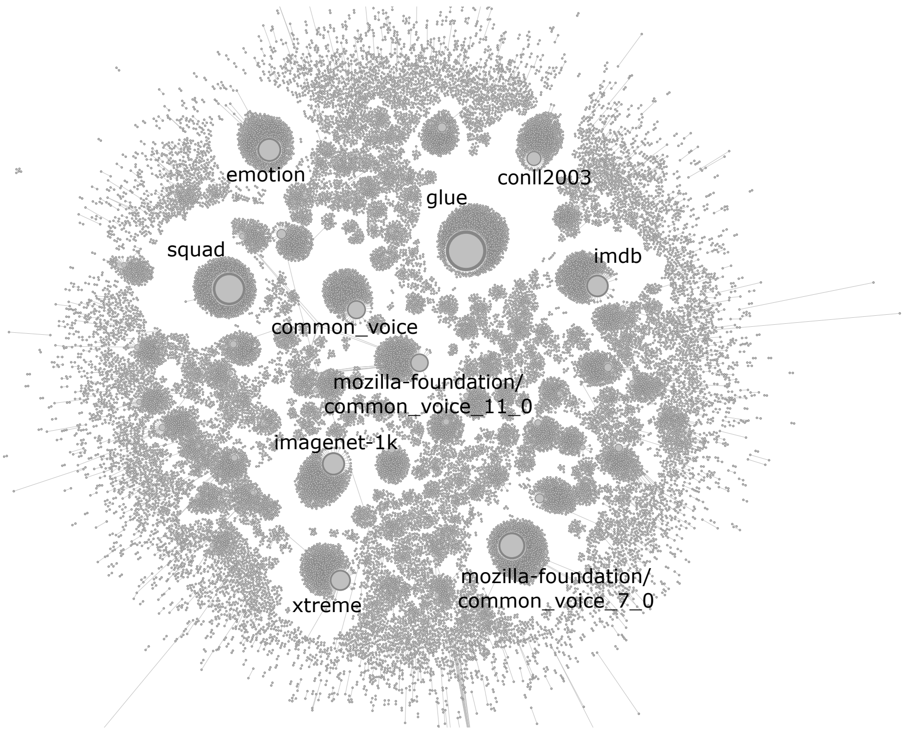
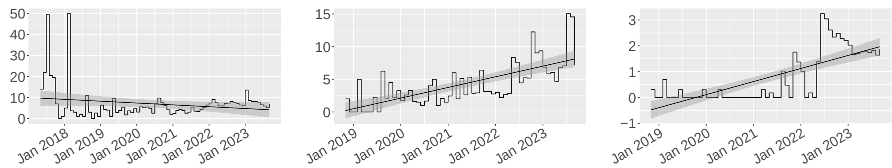


Fig. 5: Dependency graph of HFH for repositories. Only showing those with at least one dependency.

being the datasets `fka/awesome-chatgpt-prompts`, `OpenAssistant/oasst1` and `Open-Orca/OpenOrca` the most referenced with 266, 171 and 170 repositories depending on them, respectively. In the case of the top-downloaded, 40 repositories depend on others, all being models using a dataset as well. Also, 16 of the top-downloaded are used by other repositories, being datasets `glue` and `squad_v2` the most referenced with 2,027 and 410 repositories depending on them, respectively.

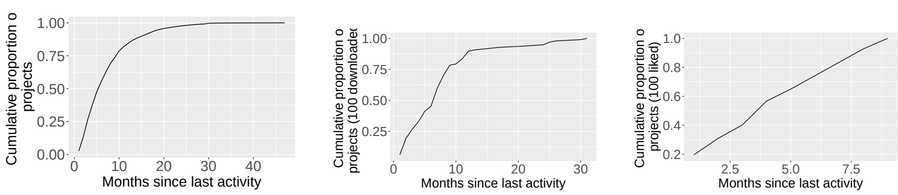
In the following, we report the **project metrics** identified in Table 8, addressing RQ2.3.

Activity. From a platform perspective, the analysis of the collection of all activity events (i.e., creation, commits, discussions, and pull requests) over time shows an exponential behavior. However, at project level, the number of events per repository does not indicate an increment of the activity on a repository (see Figure 6a). Thus, the activity growth is primarily explained by the exponential growth in the number of repositories hosted at HFH, as shown in Figure 1d. In the case of the top-100 repositories, both the top-liked and the top-downloaded shows an increment on the activity per repository (see



(a) All repositories. (b) Top-100 downloaded. (c) Top-100 liked (with logarithmic scale).

Fig. 6: Amount of activity events per repository, along with regression line and confidence interval.



(a) All repositories. (b) Top-100 downloaded. (c) Top-100 liked.

Fig. 7: Number of months without activity (since October 2023).

Figure 6b). We believe that it may be explained by the release of discussions on May 2022.

Age. Due to the exponential growth of HFH, a substantial part of repositories have been created in recent months (see Figure 1d). Further analysis revealed that, once repositories reach 1 month of life, 37.40% of them do not show any additional activity during the rest of their lifespan. To better characterize the activity over time in HFH, we also calculated the cumulative proportion of repositories since the last activity, which is shown in Figure 7. 50% of all repositories have been without any kind of activity in the last four months (and more than 60% do not have activity in the last ten months). Given this situation, we calculate the age of repositories, excluding those potentially being right-censored. Censoring is a term of survival analysis which occurs when we have information about individual survival time, but we do not know the survival time exactly (Kleinbaum and Klein, 2005). It may occur when a project has activity at the time we collected the data, but it ceases its activity later. Thus, we analyze the longevity of all repositories created six months before the timestamp of the dataset (i.e., April 2023). With this condition, we obtain a subset of 252,152 repositories, from which we observe that 85.65% of these do not have more than one month of activity. Only 4.94% surpass a lifespan of six months and 2.33% surpass one year of activity.

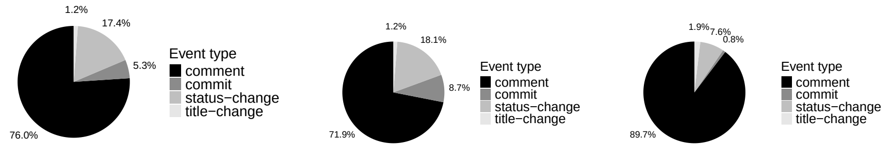
Contents. We observed that there are 67,576 repositories (10.66% of all repositories) with just one file (in fact, 67,341 of them having just the `.gitattributes` file) while only 133,791 repositories (21.11%) comprise over ten files. We also observed that 0.86% of the repositories included more than a thousand files. Most times these are dataset repositories where the whole content of the dataset has been uploaded to HFH. The resulting data distribution of the number of files per project is therefore very skewed, with a median value of 2 and IQR of $(1, 3)$ ($\mu = 172.34$, $\sigma = 2,899.79$). However, the top-100 sets are not skewed, where the top-downloaded have an average of 3.85 files ($\sigma = 3.96$) and the top-liked an average of 2.17files ($\sigma = 1.64$).

Involvement. Most repositories remain active thanks to one or two contributors ($\mu = 1.54$, $\sigma = 12.63$). When focusing on the top-100 repositories, we observe a higher user involvement, with an average of 19.08 contributors ($\sigma = 34.25$) for the top-downloaded, and an average of 180.98 contributors ($\sigma = 643.82$) for the top-liked repositories.

Interactions. The community tab is the main communication channel in a repository, which provides two types of threads: pull requests and discussions. From all repositories, only 42,578 repositories (6.70% of the total) have at least one thread, from these, 6,326 are datasets (8.07% of all datasets), 29,898 are models (7.84% of models) and 6,354 are spaces (3.60% of spaces). On average, repositories have on average 3.02 threads ($\sigma = 105.78$). On the other hand, top-100 repositories that leverage the community tab goes up to 81% of the top-downloaded and 96% in the case of the top-liked. These repositories have on average 14.23 threads ($\sigma = 24.60$) while 55.59% are pull requests for the top-downloaded, and, for the top-liked the distribution is skewed, having a median number of 49.5 threads with IQR = (23,94) ($\mu = 391.32$ and $\sigma = 2164.95$) while only 6.29% of these are pull requests. We also analyzed the distribution between pull requests and discussions. Considering all repositories, 49.91% of all threads are pull requests (77.18% in model repositories, 72.57% in datasets, and 13.58% in spaces).

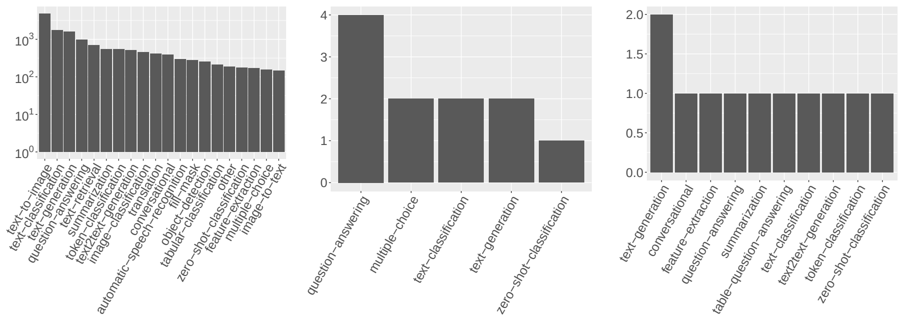
Further analysis on the events of the discussions (i.e., comments, change of status or title and commits, this last one being exclusive of pull requests), revealed that threads had a median number of 2 events. Figure 8 shows the distribution of the events in threads. The most common events are comments and commits, while change of titles and status are less frequent. However, in the case of top-downloaded repositories, the most frequent event are comments, followed by change of title status, and commits. The top-liked repositories follow a similar behavior, being the most frequent event comments, followed by change of status and title, and commits.

Artifact type. Hugging Face Hub was originally known for its NLP contributions, but its growth may allow entering other types of ML-artifacts. In this metric, we study the main ML tasks addressed by the models of HFH, and which libraries they rely on. Note that not all repositories provide this information, as it is the author's responsibility to do it via tags. We detected that only 82.32% of the repositories are tagged. Regarding ML tasks addressed (see Figure 9), only 2.48% of all repositories provide such information, being the most common `text-to-image`, `text-classification`, and `text-generation` with 4,880, 1,750 and 1,581 repositories, respectively. From the top-100 repos-



(a) All repositories. (b) Top-100 downloaded. (c) Top-100 liked.

Fig. 8: Proportion of discussion events.



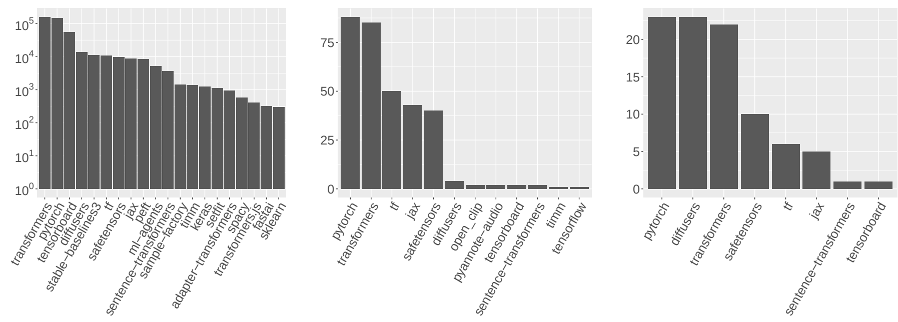
(a) All repositories (top-20, with logarithmic scale). (b) Top-100 downloaded. (c) Top-100 liked.

Fig. 9: Number of repositories addressing each task.

itories, only 11% of both the top-downloaded and the top-liked report the ML task addressed, being `question-answering` (4), `text-generation` (2, both), `multiple-choice` (2) and `text-classification` the most common. The most common libraries used are reported in Figure 10, being Transformers (158,815), PyTorch (146,235), TensorFlow (10,697), Safetensors (9.824) and JAX (8,813) the most used.[23] The most common libraries used in the top-downloaded are PyTorch (88), Transformers (85), TensorFlow (50), JAX (43) and Safetensors (40), and in the top-liked are PyTorch (23), diffusers (23), and Transformers (22). Note that `transformers.js` library is the web equivalent of the Transformers library, while `sentence-transformers` is a different library developed in the work of Reimers and Gurevych (2019).

Popularity. In order to understand the relevance of HFH repositories, we analyze the popularity statistics available (i.e., downloads and likes). These statistics are very skewed, showing how most of these are possessed by a few repositories. On one hand, the median number of downloads for a repository is 0 with IQR = (0,1) ($\mu = 1,420.78$ and $\sigma = 146,312.4$) and the median number

[23] Libraries supported by HFH: `https://github.com/huggingface/huggingface.js/blob/main/packages/tasks/src/model-libraries.ts`



(a) All repositories (top-20, with logarithmic scale).

(b) Top-100 downloaded.

(c) Top-100 liked.

Fig. 10: Libraries used in repositories.

of likes is 0 with IQR = (0,0) ($\mu = 1.11$ and $\sigma = 30.64$). On the other hand, the top-downloaded have a median number of 2,219,800 downloads with IQR = (1,451,668, 4,581,399) ($\mu = 4,713,520$ and $\sigma = 8,736,650$) and a median number of 88 likes with IQR = (35.75, 256.75) ($\mu = 345.45$ and $\sigma = 1,033.02$), and the top-liked repositories have a median number of 83,552 downloads with IQR = (13,031, 299,634) ($\mu = 1,677,634$ and $\sigma = 6,705,931$) and a median number of 1182 likes with IQR = (924.75, 1983.75) ($\mu = 1,726.76$ and $\sigma = 1,485.653$).

Answer to RQ2.1: Table 8 shows the set of validated metrics for evaluating the usage dimension in code-hosting platforms. The metrics are categorized into project and platform metrics, targeting the usage within single repositories, or at the whole platform, respectively.
Answer to RQ2.2: At platform level, HFH is under an exponential growth, hosting thousands of repositories. Despite its early stage, it provides valuable and diverse repositories, with a set of over 600k of them. One key characteristic are the dependencies between repositories, potentially building an interconnected ecosystem, highlighting relationships caused by how ML artifacts are developed (e.g., models trained with datasets hosted in HFH and showcased by spaces).
Answer to RQ2.3: At project level, the development activity is mainly comprised by Git commits and discussions. The latter seems to be more relevant when considering the top-100 repositories. Project's activity usually does not last more than a month, which might indicate that the repository is uploaded to HFH with hosting rather than with development purposes. Projects are usually maintained by one or two contributors, while in the top-100 repositories there is more participation of the community. The most addressed ML tasks are under the NLP subfield.The average repository makes use of the Transformers and PyTorch library, which allow the

Table 9: Interviewees information.

Code	Gender	Country	Experience	Role	HFH Use
R1	Male	Spain	+15 y	Senior researcher	End-user
R2	Male	Colombia	2-5 y	Junior researcher	End-user
D1	Male	Luxembourg	2-5 y	Software engineer	End-user
D2	Male	France	5-10 y	Software engineer	HFH Internal developer
D3	Male	Spain	2-5 y	Cloud & Platform Engineer	End-user

use of pre-trained models and the rapid configuration and development of ML artifacts, respectively.

5.5 Interview Results

Once analyzed HFH, we sat down with five participants for an interview. We provide basic information about the interviewees (see Table 9). For anonymity purposes, the interviewees were assigned identification codes, where R1-R2 are researchers and D1-D3 are developers. Besides personal information (i.e., gender and nationality) we report the years of experience in academia or in industry, their job profile, and their relationship with the platform. Note that for developers, we can only report the non-related research columns.

Interviewees reviewed our RQs conclusions and discussed and validated the results with us, in particular, the feature framework, as it is the result of a qualitative analysis. All of them agreed that the features identified covered the main aspects of a code-hosting platform. Moreover, R2, when presented the characterization of HFH (see Table 7), stated *"... my view of HFH is more as a data platform than a development platform."*. He perceived the focus of HFH in hosting data artifacts (i.e., providing features to search and exploit ML-artifacts) rather than to provide development features (i.e., coding topic). D3 also noted the lack of coding support, thus using GitHub for its vast support on this topic. Additionally, R1 highlighted the absence of work management (i.e., agile-like boards to organize tasks, see Table 6).

Interviewees also provided additional insights on other features. In particular, D2 helped us enrich *social profile* and *project relations* features. D2 stated that besides the traits we identified in *social profile* feature, in HFH it is also visible the authored papers and blog posts, which can potentially build a portfolio. Concerning the *project relation* feature, D2 indicated *"... it [project relations] can also be inferred automatically. So based on what we find in the model cards, we can infer without people actually tagging. Or from* *a space we check all the HTTP calls made by a space to a model and if it's downloaded then we do the link."*, which might help in the definition of these relationships, but it depends on the reliability of these automatic inferences. Finally, D2 also clarified why HFH does not give full support to coding related features, and it is more focused on social aspects: HFH is focused on providing a playground (i.e., spaces) for AI-interested users, where models can be exploited. They believe this gives more visibility to the model, thus helping users to explore the model or dataset repositories and eventually contribute, or leave feedback in the spaces' discussion threads. Because of this, even the coding features are aimed at a broader public, thus the importance of the pull request simplification, as this facilitates the contribution from non-developers users (Izquierdo and Cabot, 2022). Furthermore, R1 shared the importance of providing proper documentation to access and understand the project. R1 stated that the integration of Read the Docs[24] in GitHub has been helpful to provide autogenerated documentation for his projects. Besides the current documentation of HFH (i.e., repository card), an additional service of autogenerated documentation could be an upgrade, although he acknowledges that code documentation is not usually uploaded to HFH.

Overall, they agreed that our characterization and conclusions of RQ1 helped to provide a fair insight on the features provided by HFH and trigger some discussions on HFH specific profile, which affects the type of empirical study performed in this platform (see Section 6).

When asked about the HFH usage analysis (i.e., results for RQ2), all interviewees agreed on the exponential growth HFH is undertaking. R1 stated that the analysis provided a clear overview of HFH. D2 acknowledged the importance of the NLP community in HFH, in particular, the large amount of text-to-image projects, but stated that they are seeing an arising in other communities, for instance, NLP for local inference. R2 related the repository dependency metric (see Figure 5) to a previous study he performed in GitHub, thus indicating the chance of study replication (see Section 6).

6 Discussion

In this section, we discuss the results of our analysis, classifying them in strong and weak points of HFH from which we then derive a set of suitable (and, respectively, non-suitable) scenarios to use HFH as source for empirical studies, which is the overarching question this papers aims to answer.

6.1 Points in Favor of using HFH as source of empirical studies

Regarding the features provided by HFH, we noticed the **support for social interactions** and a friendlier interface to **all kinds of users** (i.e., not focusing on developers). As D2 stated, the intention of HFH is to provide a place

[24] `https://about.readthedocs.com/?ref=readthedocs.com`

oriented to all users, and more specifically to "AI-builders" (D2's own words), where the development workflow is envisioned as end-users playing with models or spaces, and if there is not a repository fulfilling their needs, they can create a new one (e.g., training or fine-tuning a new model, or creating a new space). They picture spaces to be the frontal page of HFH, as it is a playground where all AI enthusiasts have the chance to test the model's capabilities. Then, spaces might be exploited as a landing ground for all users, where discussions are initiated. And while some development flows are supported, the focus does not seem to be the replacing of GitHub or other code-hosting platform but, instead, complementing them.

Indeed, discussions, along with papers and blog posts, and their possible interactions (e.g., commenting on posts), facilitate users to have a **centralized place for ML related topics** (i.e., from informal conversations up to the exchange of opinions in scientific papers). D2 stated their aim to foster ML communities usually found in other social sites (e.g., Twitter). HFH provide Q&A and social profile features which allow this interaction (see Table 7). Furthermore, the different types of repositories in HFH are an important characteristic that allow dedicating a specific kind of repositories (i.e., spaces) for demonstrations and first contacts with the ML models.

Besides the features supporting social interaction, we highlight that these social features are indeed being exploited in HFH projects. In Section 5.4, we conclude that discussions are active in the top-100 repositories, specifically the top-liked. Therefore, the expected behavior of having spaces as a landing ground might be the way users will interact with the platform. Thus, spaces will work as a first interface between the development part (i.e., training the model and creating datasets) and the end-user exploitation of these ML models and datasets.

Moreover, since HFH's purpose is to **host ML-specific projects**, HFH is the ideal place to answer empirical questions on the development and interaction with ML artefacts (and/or on how these type of projects differ from other types of software development projects). In this sense, we would like to remark that HFH provides a set of exclusive features (e.g., repository type or repository dependencies among types) which deserve to be further investigated. The most present community is NLP, acknowledged by D2, but we see also a high presence of text-to-image, image classification, and other computer vision tasks. Therefore, HFH can be a promising place to study all these communities.

Another promising source of information is the defined **relationship between repositories**. As seen in Section 5.2, HFH provides support to such a feature, potentially **identifying communities** (see Figure 5) and how they interact around sets of projects of common interest.

As we have described in Section 2, HFH is following an **exponential growth** similar to the one GitHub had (see Figure 1). Thus, we might expect HFH to be considered the main hub of ML artifacts in the near future, being a potential rich source of data. Note however HFH is still in an early stage of adoption, as we discuss below (see Section 6.2).

6.2 Points Against using HFH as source of empirical studies

When addressing the suitability of HFH for empirical studies, we also have to highlight the points HFH is lacking. For instance, the intention to provide a more friendly environment to non-developer users, affects the richness of the development practices observed in other platforms, e.g., as we mention in Section 5.5, HFH provides a simplified pull request system to better engage non-developers. As we reported in Section 5.4, the increase in activity in repositories may be explained by the release of discussions. Furthermore, the **limited support to coding** (see Section 5.2) force developers to use other platforms with further coding support (e.g., GitHub) in parallel to HFH which may require some types of studies to cover both platforms. The large number of repositories being uploaded to HFH can complicate the research of ML artifacts fulfilling a particular interest. Interviewee D3 and R2 shared their experience with the platform, underlining the burden of finding the appropriate model among all available. D2 stated that their purpose is to make repositories more discoverable and facilitate the search of the ML artifacts by its traits (e.g., task or language). D1 and D3 shared that they usually choose the models based on downloads or likes. However, these approaches are far from providing an optimal selection, introducing a critical issue, the Matthew effect (i.e., commonly known as the rich get richer), where only a little fraction of the repositories hosted in HFH control the majority of these statistics and end up monopolizing the choice of the users. As we described in Section 3, other works also identified the threat of **relying on few attributes for measuring popularity**.

Inspired by the work of You et al. (2022), new ways of ranking PTMs could be integrated in HFH to index the models and ease the user experience, thus being target of new studies. But overall, there is clearly a **need for a more generic sampling strategy** that can be used for selecting a relevant subset of HFH projects depending on the empirical study needs, with the goal of maximizing the diversity of the selection and minimizing the threats to validity.

As we have seen in Section 5.4, besides the top-liked repositories, there is a **notable presence of empty and inactive repositories** (see Figures 6 and 7). In a previous work (Ait et al., 2022), we analyzed the survival rate of four ecosystems within GitHub. In this study, we noticed most repositories turn inactive in a few months after their upload, which might also be happening in HFH. While this is relevant information about the platform, studies targeting social interactions or development activities require active repositories. Thus, to overcome this threat when targeting HFH there is a need in applying sampling approaches to obtain a high-quality sample.

We are starting to see the interest of sampling strategies for GitHub, e.g., Dabic et al. (2021), where they provide a sampled dataset of the most common features targeted in MSR studies. Sometimes, we want the opposite, instead of selecting a (large) representative sample we would like to focus on a few projects that are of high-quality. What makes a project high-quality is a biased decision but an option is to ask the experts. So, in our interviews, D2 highlighted a few projects he considered especially interesting and could be used as target of individual studies. In particular, D2 mentioned `prompt-collective`, a community-produced space to label data in an open-source way.[25] with a dataset of over 10,000 prompts that can be used for training and evaluating language models on prompt ranking tasks.[26] D2 also mentioned `chatbot-arena-leaderboard` space,[27] a benchmark platform for LLMs that features anonymous, randomized battles in a crowd-sourced manner. And not just projects, some examples of active users in HFH, providing a complete social profile are `TheBloke`,[28] and `teknium`.[29] These active behavior is what HFH expects in the future, where HFH is used as a channel for promoting interesting ML projects. Thus, these repositories can be a good target in HFH of referential projects in the platform.

Furthermore, HFH is still in an early stage (see Section 2). As seen in Figure 4, most repositories have been created since 2022. Thus, the repositories' data might **not be sufficient to analyze some patterns that need longer time ranges** (e.g., annual cycles).

When collecting HFH's data, we noticed some **potential information that if available could allow for some interesting analysis**. For instance, geographical distribution could help perform studies on diversity and spread of (AI) open-source development (Wachs et al., 2022), or gender information which would allow studies on gender bias (Imtiaz et al., 2019). Moreover, we believe further indicators of usage, such as Inference API calls on models, would enrich the existing data and provide closer information about the actual use of HFH. Such data would help to understand how much are open-source models used or which ones are the most used, and if geographical data would be available which were the most common regions using the inference support.

D1, D3 and R2 shared their experiences when searching for models in HFH, stating that while the tags provided help to find task-specific projects, there is still a large amount of repositories complicating the choice. Usually, they choose a repository by looking at the current available statistics (i.e., downloads and likes), but they acknowledge they might be missing repositories that fulfill better their purposes. For instance, D1 stated that he had trouble finding a model that processed correctly the Luxembourgish. He proposed some sort of language score to help filter and select the models. This better tagging and linking (between models, datasets and spaces) could also help perform more clustering analysis of the ML-artifacts based on a number of categories and tags.

On some occasions, the data we need for an empirical study can be seen on the platform but it is **not accessible through the API**, hampering the data gathering and processing for that specific feature. For instance, HFH

[25] For more information: `https://huggingface.co/spaces/DIBT/prompt-collective`
[26] `https://huggingface.co/datasets/DIBT/10k_prompts_ranked`
[27] `https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard`
[28] `https://huggingface.co/TheBloke`
[29] `https://huggingface.co/teknium`

aims to let users indicate which articles and blog posts author, besides other personal information (e.g., personal websites or social media) but this information cannot be retrieved via the official API. Another relevant information not available through the API is the relationship between GitHub and HFH repositories, which is usually specified in the card data. Inferring this link is still possible, but it requires the analysis of the card full text with the possible errors this may generate as there is not a standard way to define this information and the card could link to more than one repository, e.g., to complement information about the model purpose or datasets. Clearly, providing this data as an API endpoint would facilitate studies targeting both platforms.

However, the **Hub API is in continuous development** and the HFH developers are receptive to new proposals and contributions, promoting community involvement, as observed in the development of their Python library.[30] We opened some issues and pull requests which were addressed promptly. Therefore, we can expect them to cover most of their features in the near future.

6.3 On the Suitable Empirical Studies on HFH

From the strong and weak points identified previously, we can discuss which empirical studies are suitable on HFH. Given the focus on establishing HFH as a more social than development platform, empirical studies on the **collaborative and networking aspects** would be ideal. As similarly done in studies analyzing discussion threads in Stack Overflow or GitHub (e.g., the work of Croft et al. (2022)), studies could be conducted on the discussion of ML topics, such as ethics, environmental, or security concerns of ML models. Furthermore, studies targeting **social aspects** could also be a good fit for the HFH. For instance, studies on the adoption of open-source software (Gwebu and Wang, 2011) or social behaviors (Yu et al., 2014) could be easily adapted.

The focus on providing a hub specific for ML artifacts, has been possible thanks to specific features, such as having specialized types of repositories (see Section 5.2). The repository types allow the exploitation of particular traits of ML artifacts (e.g., carbon footprint of ML models). This data is usually reported in the repository card, as described in Section 2, which allow **studies on specific ML concepts** such as PTM reuse, some of them already emerging as described in Section 3.

The features provided by HFH to identify dependencies between repositories, along with their effort for automatically inferring these links (see Section 5.5), allow the identification of community clusters. For instance, in Figure 5 we have observed how there are some clusters around certain repositories. Therefore, **studies in the identification of communities and graph analysis** (e.g., ecosystem health (Liao et al., 2019)) could be suitable leveraging on this feature.

[30] `https://github.com/huggingface/huggingface_hub`

HFH provides a fair range of features to be targeted in empirical studies, although some usually targeted features in GitHub studies are not provided (e.g., issues). Furthermore, while HFH provides some support to development, the usual behavior of ML projects is to upload the source code in GitHub and the artifacts (i.e., model or dataset) in HFH, also acknowledged by D2. This may change in the future (if more people move to HFH as a full platform) but this is not yet the case, and it does not seem to be the priority for HF itself.

Therefore, studies aiming at having a complete picture of the end-to-end development of an ML artefact may need to leverage **HFH in combination with GitHub or others**, rather than using HFH (or GitHub) as standalone data source.

This kind of studies might be more robust as the features HFH (partially) lacks are covered by the other platform, and vice versa, potentially introducing best-of-breed studies on ML-focused empirical studies. This is also true considering that both may attract different types of user profiles.

Furthermore, we believe that it would be interesting to **replicate existing studies** done on GitHub or other platforms. R2 stated that studies he performed in GitHub could be replicated in HFH. This leads to a discussion on the replicability in HFH of studies performed in other platforms, providing a potential area of study of distinctions in development practices performed in general-purpose platforms and in the ML-community-oriented platforms (e.g., HFH).

6.4 On the Non-suitable Empirical Studies on HFH

For almost the same reasons described in the previous section, HFH is not the best match for some other types of empirical studies, described in the following. The friendlier environment of HFH towards non-developer users such as the simplification of the pull request system, may limit and/or affect the breadth and validity of the conclusions of the study, as the target population might not be the usual developer. Then, empirical studies on **development practices** of HFH-hosted ML repositories should consider this threat, which can be mitigated by introducing new sources of data such as GitHub (see Section 6.3).

Additionally, **longitudinal or large-scale studies** (e.g., Bao et al. (2021)) on HFH are still exposed to several threats due to the recentness of HFH as aforementioned. Thus, it is not possible yet to perform such studies. However, the exponential growth of HFH might indicate the opportunity for this kind of studies in a near future.

The growth in number of repositories might be understood as a rich data source of ML artifacts, the **NLP predominance** is a threat that existing studies already noticed (see Table 2). As we report in the interview results (see Section 5.5), the internal team is aware of the NLP predominance in HFH. However, they are seeing a rising trend in other communities which might indicate this threat will be mitigated in the future, as seen in Section 5.4, such as the computer vision community. While there is no certainty, other communities might start to flourish in HFH, providing a more diverse platform. For now, this must be considered a threat.

Other limitations stem from missing data and data access as discussed in Section 6.2. For instance, the popularity of repositories is usually measured by number of downloads or likes. However, other indicators might also be relevant, such as the number of inferences made on a model. For now, as aforementioned, some information is still not available, thus hampering studies on some topics. For instance, studies on social capital Qiu et al. (2019), or online leadership (e.g., Mu et al. (2019)) could be performed if articles and blog posts published in HFH would be available via the Hub API.

7 Threats to Validity

Our work is subjected a number of threats to validity, namely: (1) internal validity, which refers to the inferences we make; (2) external validity, which is related to the generalization of our findings; (3) construction validity, which refers to the approaches we use to address the research questions; and (4) conclusion validity, which is related to the interpretation of our results.

7.1 Internal & External Validity

Regarding the internal validity, to address RQ1 we relied on our feature framework, which may not cover all the features from code-hosting platforms. The dimensions of our framework are gathered by an analysis of a set of platforms. However, these platforms provide subsets of features according to their business objectives. Furthermore, the interpretation of the features and topics is subjected to the understanding of the authors. To address RQ2, we rely on the data provided by HFCommunity, which uses data from the HFH API and Git. Git and HFH data may suffer from user clashing, as usernames in both platforms may not match, as reported by Ait et al. (2023a), which might influence in the number of users reported. It is also important to note that the emerging behavior of HFH does not demonstrate consolidation and widespread yet, which may limit the scope of our inferences.

As for the external validity, the analysis done in RQ1 relies on the current feature set of HFH at the moment of performing our study, but it may change in the future. On the other hand, the dataset used in RQ2 is based on a set of HFH projects from HFCommunity, which releases periodic versions. Thus, the results of the study should not be directly generalized without proper comparison and validation.

7.2 Construct & Conclusion Validity

The process to constructing the feature framework follows an iterative approach where each social code-hosting platform is analyzed to identify the features. In this approach, each platform is studied individually to identify its features, and then they are shared to identify a superset of features. As some features may be shared or be similar, the process repeats until no more new features are identified. In the last step, the set of features are grouped according to a topic. Before building the feature framework we perform the internal validation steps (see *Internal Validation* in Figure 2). The search query formulated in these steps identifies papers with the feature or platform in its title. However, some studies might have this information within its text rather than in the title. Furthermore, studies from other digital libraries are not considered. Similarly, the selection of metrics is based on the authors' experience in reading and performing a significant number of empirical studies, and meta-studies, paired with their experience in using also the HFH. With both the selection of metrics and identification of features, the main process is performed by the first author, and the results are debated by the second and third authors. Disagreements are discussed until a consensus is reached. To mitigate these threats, as mentioned in Section 4.2, the resulting set of features and metrics are validated by conducting a survey with both actors from the industry and from the empirical research and MSR communities. However, note that even with these mitigation efforts, some features could not have been detected or requested during the survey (e.g., platform internal integration of information sources). The low participant sample might also introduce a threat in the generalization of the conclusions.

The conclusion validity is mainly threatened by biases of our interpretation of the results of the RQs. Thus, we performed two sets of interview to mitigate this threat.

8 Conclusion

In this paper, we addressed the concern on whether HFH is suitable for performing empirical studies. For this, we have proposed a qualitative and quantitative analysis. The former aims at characterizing HFH according to a feature framework extracted from multiple code-hosting platforms. This framework could be used to characterize other ML-based hosting platforms that could appear in the future. The latter is intended to provide an insight on the current state and the data availability of HFH, proposing several metrics.

The results of our study conclude that, indeed, HFH is a very valuable source of data to better understand how the development of ML-related artifacts is done in practice. Our insights allow us to conclude that HFH focuses on social support, while also providing a rich set of data to perform empirical studies. In the discussion section, we go deeper in this conclusion and provide additional caveats and advice. Overall, we believe our results contribute to understand what kind of empirical studies can be performed in HFH.

As future work, we plan to continue monitoring the features and data available in the HFH, and to start replicating on this platform interesting empirical studies performed only on GitHub so far to compare the results. Regarding the monitoring, we plan to explore the application of metrics and principles such as the Matthew effect (Rigney, 2010), already used in the social coding world (Dabbish et al., 2012). Furthermore, we plan to develop additional tool support to facilitate the exploitation of HFH data (based on the Discussion section) including an extension to HFC that aims to link both HFH and GitHub data.

Acknowledgements This work is part of the project TED2021-130331B-I00 funded by MCIN/AEI/10.13039/501100011033 and European Union NextGenerationEU/PRTR; and BESSER, funded by the Luxembourg National Research Fund (FNR) PEARL program, grant agreement 16544475.

Data availability. The data used in RQ1 (i.e., survey results and interview transcription) is available in a Zenodo repository with the identifier `https://doi.org/10.5281/zenodo.11072131`. The data used in RQ2 is the October 2023 release of HFCommunity, available with the identifier `https://doi.org/10.5281/zenodo.10020642`.

References

Ait A, Izquierdo JLC, Cabot J (2022) An empirical study on the survival rate of github projects. In: Int. Conf. on Mining Software Repositories, pp 365–375 36

Ait A, Cánovas Izquierdo JL, Cabot J (2023a) HFCommunity: a Tool to Analyze the Hugging Face Hub Community. In: Int. Conf. on Software Analysis, Evolution and Reengineering, pp 728–732 4, 18, 40

Ait A, Izquierdo JLC, Cabot J (2023b) On the suitability of hugging face hub for empirical studies. CoRR abs/2307.14841 7

Akhtar M, Benjelloun O, Conforti C, Gijsbers P, Giner-Miguelez J, Jain N, Kuchnik M, Lhoest Q, Marcenac P, Maskey M, Mattson P, Oala L, Ruyssen P, Shinde R, Simperl E, Thomas G, Tykhonov S, Vanschoren J, van der Velde J, Vogler S, Wu C (2024) Croissant: A metadata format for ml-ready datasets. In: Workshop on Data Management for End-to-End Machine Learning, pp 1–6 6

Al-Rubaye A, Sukthankar G (2023) Improving Code Review with GitHub Issue Tracking. In: Int. Conf. on Advances in Social Networks Analysis and Mining, p 210–217 21

Alamer G, Alyahya S (2017) Open Source Software Hosting Platforms: A Collaborative Perspective's Review. J Softw 12(4):274–291 10

Baltes S, Kiefer R, Diehl S (2017) Attribution Required: Stack Overflow Code Snippets in GitHub Projects. In: Int. Conf. on Software Engineering Companion, pp 161–163 21

Baltes S, Knack J, Anastasiou D, Tymann R, Diehl S (2018) (No) Influence of Continuous Integration on the Commit Activity in GitHub Projects. In: ACM SIGSOFT Int. Workshop on Software Analytics, p 1–7 22

Bao L, Xia X, Lo D, Murphy GC (2021) A Large Scale Study of Long-Time Contributor Prediction for GitHub Projects. IEEE Trans Software Eng 47(6):1277–1298 39

Bäumer FS, Dollmann M, Geierhos M (2017) Studying Software Descriptions in SourceForge and App Stores for a Better Understanding of Real-Life Requirements. In: ACM SIGSOFT Int. Workshop on App Market Analytics, p 19–25 16

Biazzini M, Baudry B (2014) "May the Fork Be with You": Novel Metrics to Analyze Collaboration on GitHub. In: Int. Workshop on Emerging Trends in Software Metrics, p 37–43 21

Borges H, Tulio Valente M (2018) What's in a GitHub Star? Understanding Repository Starring Practices in a Social Coding Platform. J Syst Softw 146:112–129 21

Cai X, Zhu J, Shen B, Chen Y (2016) GRETA: Graph-Based Tag Assignment for GitHub Repositories. In: Annual Computer Software and Applications Conference, vol 1, pp 63–72 22

Casalnuovo C, Suchak Y, Ray B, Rubio-González C (2017) GitcProc: a tool for processing and classifying GitHub commits. In: ACM SIGSOFT Int. Symposium on Software Testing and Analysis, p 396–399 18

Castaño J, Martínez-Fernández S, Franch X, Bogner J (2023a) Analyzing the evolution and maintenance of ML models on hugging face. CoRR abs/2311.13380 6, 7

Castaño J, Martínez-Fernández S, Franch X, Bogner J (2023b) Exploring the carbon footprint of hugging face's ML models: A repository mining study. In: Int. Symposium on Empirical Software Engineering and Measurement, pp 1–12 6, 7

Chen D, Stolee KT, Menzies T (2019) Replication Can Improve Prior Results: A GitHub Study of Pull Request Acceptance. In: Int. Conf. on Program Comprehension, p 179–190 21

Cosentino V, Cánovas Izquierdo JL, Cabot J (2016) Findings from GitHub: Methods, Datasets and Limitations. In: Int. Conf. on Mining Software Repositories, p 137–141 17

Cosentino V, Cánovas Izquierdo JL, Cabot J (2017) A Systematic Mapping Study of Software Development with GitHub. IEEE Access 5:7173–7192 5, 12

Croft R, Xie Y, Zahedi M, Babar MA, Treude C (2022) An empirical study of developers' discussions about security challenges of different programming languages. Empir Softw Eng 27(1):27 38

Dabbish LA, Stuart HC, Tsay J, Herbsleb JD (2012) Social coding in github: transparency and collaboration in an open software repository. In: Conf. on

Computer Supported Cooperative Work, pp 1277–1286 42

Dabic O, Aghajani E, Bavota G (2021) Sampling projects in github for MSR studies. In: Int. Conf. on Mining Software Repositories, IEEE, pp 560–564 5, 36

Decan A, Mens T, Claes M, Grosjean P (2016) When GitHub Meets CRAN: An Analysis of Inter-Repository Package Dependency Problems. In: Int. Conf. on Software Analysis, Evolution, and Reengineering, pp 493–504 21

Demeyer S, Murgia A, Wyckmans K, Lamkanfi A (2013) Happy Birthday! a Trend Analysis on Past Msr Papers. In: Int. Working Conf. on Mining Software Repositories, pp 353–362 5

Destefanis G, Ortu M, Bowes D, Marchesi M, Tonelli R (2018) On Measuring Affects of Github Issues' Commenters. In: Int. Workshop on Emotion Awareness in Software Engineering, p 14–19 21

Dyer R, Nguyen HA, Rajan H, Nguyen TN (2015) Boa: Ultra-Large-Scale Software Repository and Source-Code Mining. ACM Trans Softw Eng Methodol 25(1) 18

Eibl G, Thurnay L (2023) The Promises and Perils of Open Source Software Release and Usage by Government – Evidence from GitHub and Literature. In: Int. Conf. on Digital Government Research, p 180–190 21

English R, Schweik CM (2007) Identifying Success and Tragedy of FLOSS Commons: A Preliminary Classification of Sourceforge.net Projects. In: Int. Workshop on Emerging Trends in FLOSS Research and Development, pp 11–11 16

Eraslan S, Kopec-Harding K, Jay C, Embury SM, Haines R, Cortés Ríos JC, Crowther P (2020) Integrating GitLab metrics into coursework consultation sessions in a software engineering course. J Syst Softw 167:110613 16

Fairbanks J, Tharigonda A, Eisty NU (2023) Analyzing the Effects of CI/CD on Open Source Repositories in GitHub and GitLab. In: Int. Conf. on Software Engineering Research, Management and Applications, pp 176–181 16

Flint SW, Chauhan J, Dyer R (2022) Pitfalls and Guidelines for Using Time-based Git Data. Empir Softw Eng 27(7):194 6

Foushee B, Krein JL, Wu J, Buck R, Knutson CD, Pratt LJ, MacLean AC (2013) Reflexivity, Raymond, and the Success of Open Source Software Development: A SourceForge Empirical Study. In: Int. Conf. on Evaluation and Assessment in Software Engineering, p 246–251 16

Gajanayake R, Hiras M, Gunathunga P, Janith Suºpun E, Karunasenna A, Bandara P (2020) Candidate Selection for the Interview using GitHub Profile and User Analysis for the Position of Software Engineer. In: Int. Conf. on Advancements in Computing, pp 168–173 20

Giner-Miguelez J, Gómez A, Cabot J (2024) Describeml: A dataset description tool for machine learning. Sci Comput Program 231:103030 6

Golzadeh M, Decan A, Legay D, Mens T (2021) A ground-truth dataset and classification model for detecting bots in GitHub issue and PR comments. J Syst Softw 175:110911 17

Gonzalez D, Zimmermann T, Nagappan N (2020) The State of the ML-universe: 10 Years of Artificial Intelligence & Machine Learning Software

Development on GitHub. In: Int. Conf. on Mining Software Repositories, pp 431–442 7
Gousios G, Spinellis D (2012) GHTorrent: Github's data from a firehose. In: Working Conf. of Mining Software Repositories, pp 12–21 18
Gousios G, Pinzger M, van Deursen A (2014) An exploratory study of the pull-based software development model. In: Int. Conf. on Software Engineering, pp 345–355 21
Gwebu KL, Wang J (2011) Adoption of Open Source Software: The role of social identification. Decis Support Syst 51:220–229 38
Hauff C, Gousios G (2015) Matching GitHub developer profiles to job advertisements. In: Working Conf. on Mining Software Repositories, p 362–366 20
He R, He H, Zhang Y, Zhou M (2023) Automating Dependency Updates in Practice: An Exploratory Study on GitHub Dependabot. IEEE Trans Softw Eng 49(8):4004–4022 22
Hove SE, Anda B (2005) Experiences from Conducting Semi-structured Interviews in Empirical Software Engineering Research. In: Int. Symposium on Software Metrics, p 23 14
Howison J, Crowston K (2004) The Perils and Pitfalls of Mining Sourceforge. In: Int. Workshop on Mining Software Repositories, pp 7–11 6
Imtiaz N, Middleton J, Chakraborty J, Robson N, Bai GR, Murphy-Hill ER (2019) Investigating the effects of gender bias on GitHub. In: Int. Conf. on Software Engineering, pp 700–711 37
Izquierdo JLC, Cabot J (2022) On the analysis of non-coding roles in open source development. Empir Softw Eng 27(1):18 34
Jiang W, Cheung C, Thiruvathukal GK, Davis JC (2023a) Exploring Naming Conventions (and Defects) of Pre-trained Deep Learning Models in Hugging Face and Other Model Hubs. CoRR abs/2310.01642 6, 7
Jiang W, Synovic N, Hyatt M, Schorlemmer TR, Sethi R, Lu YH, Thiruvathukal GK, Davis JC (2023b) An Empirical Study of Pre-Trained Model Reuse in the Hugging Face Deep Learning Model Registry. In: Int. Conf. on Software Engineering, p 2463–2475 5, 6, 7, 16, 18, 25
Joshi A, Kale S, Chandel S, Pal DK (2015) Likert scale: Explored and explained. British journal of applied science & technology 7(4):396–403 18
Joshi SD, Chimalakonda S (2019) RapidRelease: A Dataset of Projects and Issues on Github with Rapid Releases. In: Int. Conf. on Mining Software Repositories, p 587–591 17
Kaide K, Tamada H (2022) Argo: Projects' Time-Series Data Fetching and Visualizing Tool for GitHub. In: Int. Summer Virtual Conf. on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, pp 141–147 18
Kalliamvakou E, Gousios G, Blincoe K, Singer L, Germán DM, Damian DE (2014) The Promises and Perils of Mining GitHub. In: Int. Working Conf. on Mining Software Repositories, pp 92–101 6
Kalliamvakou E, Gousios G, Blincoe K, Singer L, Germán DM, Damian DE (2016) An In-depth Study of the Promises and Perils of Mining GitHub.

Empir Softw Eng 21(5):2035–2071 6

Kathikar A, Nair A, Lazarine B, Sachdeva A, Samtani S (2023) Assessing the vulnerabilities of the open-source artificial intelligence (AI) landscape: A large-scale analysis of the hugging face platform. In: Int. Conf. on Intelligence and Security Informatics, pp 1–6 5, 7

Kleinbaum DG, Klein M (2005) Survival Analysis: A Self-Learning Text. Springer Science and Business Media, LLC 29

Kritikos A, Chatziasimidis F (2011) SFparser: A Tool for Selectively Parsing SourceForge. In: Panhellenic Conf. on Informatics, pp 161–165 18

Lazarine B, Zhang Z, Sachdeva A, Samtani S, Zhu H (2022) Exploring the Propagation of Vulnerabilities from GitHub Repositories Hosted by Major Technology Organizations. In: Workshop on Cyber Security Experimentation and Test, p 145–150 21

Liao Z, Yi M, Wang Y, Liu S, Liu H, Zhang Y, Zhou Y (2019) Healthy or not: A way to predict ecosystem health in github. Symmetry 11(2):144 38

Malan DJ (2022) Standardizing Students' Programming Environments with Docker Containers: Using Visual Studio Code in the Cloud with GitHub Codespaces. In: ACM Conf. on Innovation and Technology in Computer Science Education, p 599–600 21

Mitchell M, Wu S, Zaldivar A, Barnes P, Vasserman L, Hutchinson B, Spitzer E, Raji ID, Gebru T (2019) Model cards for model reporting. In: Conf. on Fairness, Accountability, and Transparency, pp 220–229 3

Montandon JE, Valente MT, Silva LL (2021) Mining the Technical Roles of GitHub Users. Inf Softw Technol 131:106485 21

Mu W, Bian Y, Zhao JL (2019) The role of online leadership in open collaborative innovation. Ind Manag Data Syst 119(9):1969–1987 40

Pina D, Goldman A, Seaman C (2022) Sonarlizer xplorer: a tool to mine github projects and identify technical debt items using SonarQube. In: Int. Conf. on Technical Debt, p 71–75 18

Qiu HS, Nolte A, Brown A, Serebrenik A, Vasilescu B (2019) Going farther together: the impact of social capital on sustained participation in open source. In: Int. Conf. on Software Engineering, pp 688–699 40

Reimers N, Gurevych I (2019) Sentence-bert: Sentence embeddings using siamese bert-networks. In: Conf. on Empirical Methods in Natural Language Processing, pp 3980–3990 31

Ren L, Zhou S, Kästner C (2018) Forks Insight: Providing an Overview of GitHub Forks. In: Int. Conf. on Software Engineering: Companion Proceeedings, p 179–180 21

Rigney D (2010) The Matthew effect: How advantage begets further advantage. Columbia University Press 42

Robles G (2010) Replicating MSR: a Study of the Potential Replicability of Papers Published in the Mining Software Repositories Proceedings. In: Int. Working Conf. on Mining Software Repositories, pp 171–180 7

Robles G, Ho-Quang T, Hebig R, Chaudron MRV, Fernandez MA (2017) An Extensive Dataset of UML Models in GitHub. In: Int. Conf. on Mining Software Repositories, p 519–522 17

Romano S, Caulo M, Buompastore M, Guerra L, Mounsif A, Telesca M, Baldassarre MT, Scanniello G (2021) G-Repo: a Tool to Support MSR Studies on GitHub. In: Int. Conf. on Software Analysis, Evolution and Reengineering, pp 551–555 18

Safari H, Sabri N, Shahsavan F, Bahrak B (2020) An Analysis of GitLab's Users and Projects Networks. In: Int. Symposium onTelecommunications, pp 194–200 16

Sanh V, Wolf T, Ruder S (2019) A Hierarchical Multi-Task Approach for Learning Embeddings from Semantic Tasks. In: Conf. on Artificial Intelligence, pp 6949–6956 3

Souza I, Campello L, Rodrigues E, Guedes G, Bernardino M (2021) An Analysis of Automated Code Inspection Tools for Php Available on GitHub Marketplace. In: Symp. on Systematic and Automated Software, pp 10–17 22

Spinellis D, Kotti Z, Mockus A (2020) A Dataset for GitHub Repository Deduplication. In: Int. Conf. on Mining Software Repositories, p 523–527 17

Squire M (2017) The Lives and Deaths of Open Source Code Forges. In: Int. Symposium on Open Collaboration, OpenSym, pp 15:1–15:8 4, 5

Tsay J, Dabbish L, Herbsleb J (2014) Let's Talk about It: Evaluating Contributions through Discussion in GitHub. In: ACM SIGSOFT Int. Symposium on Foundations of Software Engineering, p 144–154 21

Valenzuela-Toledo P, Bergel A, Kehrer T, Nierstrasz O (2023) EGAD: A moldable tool for GitHub Action analysis. In: Int. Conf. on Mining Software Repositories, pp 260–264 18

Wachs J, Nitecki M, Schueller W, Polleres A (2022) The Geography of Open Source Software: Evidence from GitHub. Technological Forecasting and Social Change 176:121478 37

Wang J, Zhang X, Chen L, Xie X (2022) Personalizing label prediction for GitHub issues. Information and Software Technology 145:106845 22

Wessel M, Serebrenik A, Wiese I, Steinmacher I, Gerosa MA (2020) What to Expect from Code Review Bots on GitHub? A Survey with OSS Maintainers. In: Brazilian Symposium on Software Engineering, p 457–462 21

Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B (2012) Experimentation in Software Engineering. Springer 7, 13, 14

Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, Cistac P, Rault T, Louf R, Funtowicz M, Davison J, Shleifer S, von Platen P, Ma C, Jernite Y, Plu J, Xu C, Scao TL, Gugger S, Drame M, Lhoest Q, Rush AM (2020) Transformers: State-of-the-Art Natural Language Processing. In: Conf. on Empirical Methods in Natural Language Processing, pp 38–45 3

Wolter T, Barcomb A, Riehle D, Harutyunyan N (2023) Open Source License Inconsistencies on GitHub. ACM Trans Softw Eng Methodol 32(5) 22

Wu J, He H, Xiao W, Gao K, Zhou M (2022) Demystifying Software Release Note Issues on GitHub. In: Int. Conf. on Program Comprehension, p 602–613 21

Yang X, Liang W, Zou J (2024) Navigating dataset documentations in AI: A large-scale analysis of dataset cards on hugging face. CoRR abs/2401.13822 6, 7

You K, Liu Y, Zhang Z, Wang J, Jordan MI, Long M (2022) Ranking and tuning pre-trained models: A new paradigm for exploiting model hubs. J Mach Learn Res 23:209:1–209:47 36

Yu Y, Yin G, Wang H, Wang T (2014) Exploring the patterns of social behavior in GitHub. In: Int. Workshop on Crowd-based Software Development Methods and Technologies, pp 31–36 38

Yu Y, Wang H, Filkov V, Devanbu P, Vasilescu B (2015) Wait for It: Determinants of Pull Request Evaluation Latency on GitHub. In: Working Conf. on Mining Software Repositories, p 367–371 21

Yu Y, Li Z, Yin G, Wang T, Wang H (2018) A Dataset of Duplicate Pull-Requests in Github. In: Int. Conf. on Mining Software Repositories, p 22–25 17

Zou W, Zhang W, Xia X, Holmes R, Chen Z (2019) Branch Use in Practice: A Large-Scale Empirical Study of 2,923 Projects on GitHub. In: Int. Conf. on Software Quality, Reliability and Security, pp 306–317 21

Özçevik Y, Altay O (2023) MetricHunter: A software metric dataset generator utilizing SourceMonitor upon public GitHub repositories. SoftwareX 23:101499 17

Appendix

8.1 Queries of Digital Libraries

8.1.1 Review of platform studies

To check how many articles are published for each platform, we queried each digital library with each one of the ten platforms identified in Section 4.2.1.[a]

[a] Platforms with names with a possible space separator have been searched with both forms (e.g., GitHub or Git Hub, HuggingFace or Hugging Face, etc.)

Listing 1: ACM DL query

```
[title:platform]
```

Listing 2: IEEE Xplore query

```
("Document Title":platform)
```

Listing 3: Science Direct query

```
Title: platform
```

8.1.2 Review of literature

In order to show a generalization of the queries, we show the structure we follow, in which the `platform` keyword is one of four code-hosting platforms identified with results in Section 5.1 (*Review of literature* step), and the `feature` keyword can be one of the following values: `Branches`, `CICD - Development Workflow`, `Collaboration/Cloud Coding`,[b] `Code Review`, `CVS`, `External Integrations`, `Following`, `Fork`, `Groups`, `Issues`, `Licensing`, `Marketplace`, `Packages`, `Pull Request`, `Project Relations`, `Q&A`, `Release`, `Repo Type`, `Roles`, `Security`, `Snippets`, `Stream Analytics`, `Tagging`, `Webhooks`, `Wiki`, `Work Management`,[c] `Web Publish`.[d]

[b] In GitHub we used the keyword `Codespaces`.
[c] In GitHub we used the keyword `GitHub Projects`.
[d] In GitHub and GitLab we used the keyword `Pages`.

Listing 4: ACM DL query

```
[title:platform] AND [title:feature]
```

Listing 5: IEEE Xplore query

```
("Document Title":platform) AND
("Document Title":feature)
```

Listing 6: Science Direct query

```
Title: platform AND Title: feature
```

8.1.3 Review of datasets

The `platform` keyword is one of the four code-hosting platforms identified with results in Section 5.1 (*Review of literature* step) and the `data` keyword is either: `data source`, `dataset` or `tool`.

Listing 7: ACM DL query

```
[title:platform] AND [title:data]
```

Listing 8: IEEE Xplore query

```
("Document Title":platform) AND
("Document Title":data)
```

Listing 9: Science Direct query

```
Title: platform AND Title: data
```

8.2 Top-100 repositories

Table 10: Top-100 downloaded repositories of HFH (October 2023).

Repository type	Repository	# downloads
model	jonatasgrosman/wav2vec2-large-xlsr-53-english	71213786
model	bert-base-uncased	45579537
model	gpt2	25509863
model	NousResearch/Llama-2-13b-hf	17474551
model	xlm-roberta-base	12779762
model	openai/clip-vit-large-patch14	10691131
model	MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli	10304597
dataset	argilla/databricks-dolly-15k-curated-en	10039207
model	distilbert-base-uncased-finetuned-sst-2-english	9947348
model	runwayml/stable-diffusion-v1-5	8869292
model	sentence-transformers/all-mpnet-base-v2	8721346
model	openai/clip-vit-base-patch32	8298028
model	benjamin/wtp-canine-s-1l	8244939
model	distilbert-base-uncased	7869188
model	roberta-base	7149919
model	albert-base-v2	6608487
model	bert-base-cased	6101067
model	cardiffnlp/twitter-roberta-base-irony	5430690
model	SamLowe/roberta-base-go_emotions	5261196
model	stabilityai/stable-diffusion-2-1	5168296
model	marieke93/MiniLM-evidence-types	5094124
model	Ashishkr/query_wellformedness_score	5090057
model	microsoft/deberta-base	5078837
model	microsoft/layoutlmv3-base	5053165
model	CompVis/stable-diffusion-safety-checker	4895258
model	google/flan-t5-base	4476779
model	t5-small	4449451
model	salesken/query_wellformedness_score	4431324
model	distilbert-base-multilingual-cased	4380341
model	stabilityai/stable-diffusion-xl-base-1.0	3998348
model	stabilityai/StableBeluga-7B	3938561
dataset	squad_v2	3873236
model	google/electra-base-discriminator	3643563
model	xlm-roberta-large	3564979
model	prajjwal1/bert-small	3554508
model	roberta-large	3546690
model	camembert-base	3406542
model	google/vit-base-patch16-224-in21k	3347035
model	distilroberta-base	3053573
model	stabilityai/stable-diffusion-xl-refiner-1.0	2922286
model	sentence-transformers/all-MiniLM-L6-v2	2801660
model	allenai/longformer-base-4096	2798017
model	facebook/bart-large-mnli	2715804
model	google/flan-t5-large	2579893
model	t5-base	2445340
model	j-hartmann/emotion-english-distilroberta-base	2392056
model	nlpconnect/vit-gpt2-image-captioning	2325458
model	bert-base-multilingual-cased	2320968
model	yiyanghkust/finbert-tone	2284555
model	pyannote/segmentation	2239918
model	timm/resnet50.a1_in1k	2199681
model	alimazhar-110/website_classification	2166622
model	jonatasgrosman/wav2vec2-large-xlsr-53-russian	2142134
model	ProsusAI/finbert	2105342
model	cl-tohoku/bert-base-japanese	2048630
model	pyannote/speaker-diarization	2027372
model	patrickjohncyh/fashion-clip	1945234
dataset	tasksource/bigbench	1892966
model	facebook/bart-large-cnn	1892463
model	distilgpt2	1840862
model	nateraw/vit-age-classifier	1758995
dataset	truthful_qa	1700402
model	microsoft/resnet-50	1686914
model	facebook/wav2vec2-base-960h	1636140
model	prajjwal1/bert-tiny	1624653
model	YituTech/conv-bert-base	1599015
model	google/fnet-base	1584257
model	openai/clip-vit-base-patch16	1546491
model	deepset/roberta-base-squad2	1519436
model	microsoft/layoutlmv2-base-uncased	1492789
model	allenai/scibert_scivocab_uncased	1476644
model	alexandrainst/scandi-nli-large	1476232
dataset	cais/mmlu	1461538
model	nlptown/bert-base-multilingual-uncased-sentiment	1453836
model	martin-ha/toxic-comment-model	1452226
model	joeddav/xlm-roberta-large-xnli	1449993
model	cardiffnlp/twitter-roberta-base-sentiment	1409756
model	laion/CLIP-ViT-B-32-laion2B-s34B-b79K	1379866
model	cardiffnlp/twitter-roberta-base-sentiment-latest	1365659
model	Intel/dpt-hybrid-midas	1315741
model	Salesforce/codet5-base	1313436
model	dbmdz/bert-large-cased-finetuned-conll03-english	1281093
dataset	glue	1275020
model	ckiplab/bert-base-chinese-ner	1260619
model	laion/CLIP-ViT-H-14-laion2B-s32B-b79K	1229721
model	bigscience/bloom-560m	1222650
model	microsoft/layoutlm-base-uncased	1199626
model	lengyue233/content-vec-best	1190009
model	Riiid/sheep-duck-llama-2	1175902
model	meta-llama/Llama-2-7b-chat-hf	1166567
model	google/bert_uncased_L-2_H-128_A-2	1142908
model	setu4993/LEALLA-small	1128140
model	vinai/xphonebert-base	1091090
model	dslim/bert-base-NER	1053805
model	bert-large-uncased	1051887
model	facebook/opt-125m	1040830
model	cardiffnlp/twitter-xlm-roberta-base-sentiment	1008696
model	microsoft/wavlm-large	1002116
model	hf-internal-testing/tiny-random-gpt2	950852
model	distilbert-base-uncased-distilled-squad	950570

Table 11: Top-100 liked repositories of HFH (October 2023).

Repository type	Repository	# likes
model	runwayml/stable-diffusion-v1-5	9347
space	stabilityai/stable-diffusion	9160
model	CompVis/stable-diffusion-v1-4	5966
space	HuggingFaceH4/open_llm_leaderboard	5567
space	dalle-mini/dalle-mini	5246
model	bigscience/bloom	4036
dataset	fka/awesome-chatgpt-prompts	3529
model	WarriorMama777/OrangeMixs	3411
model	lllyasviel/ControlNet	3218
model	stabilityai/stable-diffusion-2-1	3206
model	stabilityai/stable-diffusion-xl-base-1.0	2994
model	prompthero/openjourney	2910
space	facebook/MusicGen	2815
model	meta-llama/Llama-2-7b	2684
model	THUDM/chatglm-6b	2620
space	JavaFXpert/Chat-GPT-LangChain	2546
model	CompVis/stable-diffusion-v-1-4-original	2539
model	lllyasviel/ControlNet-v1-1	2498
model	bigcode/starcoder	2357
model	tiiuae/falcon-40b	2291
model	hakurei/waifu-diffusion	2256
model	andite/anything-v4.0	2172
space	jbilcke-hf/ai-comic-factory	2149
space	pharmapsychotic/CLIP-Interrogator	2078
space	microsoft/HuggingGPT	2040
space	pharma/CLIP-Interrogator	1965
space	AP123/IllusionDiffusion	1927
model	databricks/dolly-v2-12b	1869
model	THUDM/chatglm2-6b	1784
model	stabilityai/stable-diffusion-2	1640
model	dreamlike-art/dreamlike-photoreal-2.0	1533
space	DeepFloyd/IF	1481
space	Gustavosta/MagicPrompt-Stable-Diffusion	1478
model	gsdf/Counterfeit-V2.5	1456
space	camenduru-com/webui	1418
model	meta-llama/Llama-2-70b-chat-hf	1410
model	gpt2	1404
space	suno/bark	1350
model	meta-llama/Llama-2-7b-chat-hf	1350
model	stabilityai/stable-diffusion-xl-base-0.9	1330
model	runwayml/stable-diffusion-inpainting	1318
model	webui/ControlNet-modules-safetensors	1309
model	EleutherAI/gpt-j-6b	1286
space	ysharma/ChatGPT4	1282
space	sanchit-gandhi/whisper-jax	1281
space	CompVis/stable-diffusion-license	1251
space	damo-vilab/modelscope-text-to-video-synthesis	1248
model	decapoda-research/llama-7b-hf	1229
space	huggingface-projects/QR-code-AI-art-generator	1222
space	camenduru/webui	1213
model	openai/whisper-large-v2	1151
model	prompthero/openjourney-v4	1146
model	bert-base-uncased	1125
space	timbrooks/instruct-pix2pix	1102
model	tiiuae/falcon-40b-instruct	1096
space	akhaliq/AnimeGANv2	1095
model	stabilityai/sd-vae-ft-mse-original	1093
model	mosaicml/mpt-7b	1079
space	ysharma/Explore_llamav2_with_TGI	1072
dataset	gsdf/EasyNegative	1055
space	stabilityai/stablelm-tuned-alpha-chat	1049
space	mteb/leaderboard	1042
model	andite/pastel-mix	1039
dataset	OpenAssistant/oasst1	1038
model	hakurei/waifu-diffusion-v1-4	1030
space	togethercomputer/OpenChatKit	1015
space	anzorq/finetuned_diffusion	994
model	dreamlike-art/dreamlike-diffusion-1.0	991
space	fffiloni/img-to-music	985
model	sentence-transformers/all-MiniLM-L6-v2	976
space	openai/whisper	965
model	nuigurumi/basil_mix	957
model	OpenAssistant/oasst-sft-6-llama-30b-xor	947
space	hysts/ControlNet	928
model	google/flan-t5-xxl	926
model	stabilityai/stable-diffusion-xl-refiner-1.0	921
model	monster-labs/control_v1p_sd15_qrcode_monster	913
model	Envvi/Inkpunk-Diffusion	906
dataset	Nerfgun3/bad_prompt	905
space	sczhou/CodeFormer	902
model	CompVis/stable-diffusion	902
model	nitrosocke/mo-di-diffusion	902
model	microsoft/phi-1_5	892
space	DragGan/DragGan	881
model	tiiuae/falcon-7b	878
dataset	togethercomputer/RedPajama-Data-1T	876
space	huggingface-projects/diffuse-the-rest	859
model	wavymulder/Analog-Diffusion	846
model	mistralai/Mistral-7B-v0.1	832
model	stabilityai/StableBeluga2	828
space	fffiloni/CLIP-Interrogator-2	828
space	JohnSmith9982/ChuanhuChatGPT	816
model	tiiuae/falcon-180B	811
space	Logspace/Langflow	786
space	hysts/ControlNet-v1-1	786
model	baichuan-inc/Baichuan-7B	784
space	tiiuae/falcon-180b-demo	781
model	Lykon/DreamShaper	766
space	Vision-CAIR/minigpt4	763
dataset	Open-Orca/OpenOrca	747