Information Systems Frontiers
Manuscript No. (will be inserted by the editor)

On the Popularity of Internet of Things Projects in Online Communities

An Empirical Study of Hackster.io

Taher Ahmed Ghaleb · Daniel Alencar da Costa · Ying Zou

Author pre-print copy. The final publication is available at Springer via:
http://dx.doi.org/10.1007/s10796-021-10157-1

Abstract Online Internet of Things (IoT) communities allow IoT engineers to publish information about their projects to a wider audience of users. Despite the growing adoption of IoT technologies in business, the popularity of IoT projects remains unexplored. Understanding how to improve the popularity of IoT projects helps project owners attract more users and foster business opportunities. In this paper, we explore the important characteristics of popular IoT projects across three facets: *views* count, *respects* count, and *trending* scores. We study over $18,000$ IoT projects hosted on Hackster.io—a large online IoT community. In particular, we perform a time-series clustering to identify the evolution of each of the three popularity facets. In addition, we construct linear mixed-effect models to investigate the most important factors associated with the popularity of IoT projects. We provide insights to online IoT communities to improve the user guidelines to help (new) IoT engineers make their projects more eye-catching.

Keywords Online communities; Popularity; Internet of Things (IoT); Empirical study

Taher Ahmed Ghaleb
School of Computing, Queen's University, Kingston, Canada.
E-mail: taher.ghaleb@queensu.ca

Daniel Alencar da Costa
Department of Information Science, University of Otago, Dunedin, New Zealand. E-mail: danielcalencar@otago.ac.nz

Ying Zou
Department of Electrical and Computer Engineering, Queen's University, Kingston, Canada. E-mail: ying.zou@queensu.ca

1 Introduction

The Internet of Things (IoT) is transforming human life at an unprecedented rate. Users can perform diverse tasks benefiting from the IoT technology. For instance, users can remotely control the house temperature using IoT applications. According to the International Data Corporation (IDC),[1] there will be a market size of $1.1 trillion with billions of devices connected throughout the IoT ecosystem by 2023. However, the heterogeneity and immature standardization of IoT systems increase the complexity of developing IoT systems [10, 34]. Compared to desktop/server or mobile systems, IoT systems involve different types of devices, data exchange protocols, and deployment environments. In addition, developing IoT projects requires knowledge of different hardware platforms (e.g., *Arduino*,[2] *Raspberry Pi*,[3] and *SparkFun*[4]) or cloud platforms (e.g., Amazon Web Service (AWS) IoT[5] and Microsoft Azure[6]). Furthermore, IoT practitioners need to acquire knowledge of certain domains, such as signal processing [11].

Recently, online IoT communities have become popular among IoT practitioners as an increasing number of IoT engineers publish their projects online. Online IoT communities (e.g., Hackster.io[7] and Instructables[8]) allow IoT practitioners to gain and share knowledge with each other about the latest IoT technology [51].

[1] https://www.idc.com/getdoc.jsp?containerId=IDC_P24793
[2] https://www.hackster.io/arduino
[3] https://www.hackster.io/raspberry-pi
[4] https://www.hackster.io/sparkfun
[5] https://aws.amazon.com/iot
[6] https://azure.microsoft.com
[7] https://hackster.io
[8] https://www.instructables.com

As online IoT communities grow and serve a broader audience, they allow IoT engineers and professionals to create business opportunities, especially for those who have more popular projects than other users. For example, the web page of a popular project can serve as an advertiser for a company that supports that project. Besides, the owners of popular IoT projects can earn a reputation whenever one of their projects is viewed, respected, or featured by the IoT community.[9] Such reputation can be exchanged with products in online IoT stores.[10]

Prior research has studied the factors that share a significant association with the popularity of open source software projects in online communities for open source and mobile applications [12,29,52,57]. For example, Tian et al. [57] studied the factors that are significantly associated with highly rated free Android applications. Tian et al. observed that the number of promotional images displayed by an app on the Google Play Store and the size of the app are strongly correlated with highly rated Android apps [39]. Despite the similarities between IoT communities and open source and mobile communities (e.g., in terms of online discussions and user feedback), communities of IoT projects have unique characteristics and issues are still unexplored. IoT projects rely heavily on the the hardware in addition to the software, which denotes that users require time, effort, and money to obtain and assemble the IoT devices to replicate a project. In addition, IoT communities organize contests in which IoT projects compete to address certain real-world problems. Therefore, it is important to investigate how active are online IoT communities and understand the factors that help IoT engineers promote their projects to gain more popularity.

In this paper, we study the factors that are associated with the popularity of IoT projects across three popularity measures: *views count*, *respects count*, and *trending score*. The *views count* captures the number of times a project is viewed by the community users. The *respects count* captures the number of thumbs-ups a project has received from the community users. The *trending score* captures the order of a project among other projects considering the age of the project, views count, and respects count. *Hackster.io* maintains a dedicated list to allow users to show projects based on their views count, respects count, or trending. Hence, it is important for project owners to understand what makes their projects successful in each of such popularity measures. Moreover, a higher number of views may indicate that a project has reached more users, which could be due to the use of advertisements. While making a project more reachable is important, project owners still desire to gain a positive user perception. Hence, in this study, we aim to generate insights into not only what makes a project more reachable but also what makes a project more perceivable by the IoT community users. Previous research by Borges et al. [8] surveyed developers and found that, while the majority of developers care more about project stars, still over two-thirds of developers also care about the number of project watchers and forks. Hence, it is important to study each of the popularity measures independently to generate insights into the factors that project owners can tweak to improve their projects with respect to views, respects, and trending scores.

To investigate the popularity of IoT projects, we conduct an exploratory study using online communities. We investigate over $18,000$ IoT projects that are hosted on *Hackster.io* (referred to as Hackster from now on). Hackster stands out from other online communities because it has a large user base (i.e., over 1.6 million users as of today). Hackster is sponsored by Microsoft, Intel, Google, and Amazon. In addition, IoT projects hosted on Hackster can be developed and deployed using IoT cloud services, such as Amazon AWS and Microsoft Azure. Such projects comprehend a wide diversity of IoT projects, such as home automation (e.g., controlling home surveillance cameras and air conditioners remotely).

Paper organization. The rest of this paper is organized as follows. Section 3 introduces the experimental setup of our empirical study. Section 4 discusses the results and findings of our studied RQs. Section 5 discusses the implications of our findings for project owners and the Hackster community. Section 6 describes threats to the validity of our results. Section 7 presents the related literature on IoT technology and the popularity of software projects. Finally, Section 8 concludes the paper and outlines avenues for future work.

2 Background - the Hackster IoT community

Hackster is an online community dedicated to both (a) beginners to learn about hardware IoT development and (b) professionals to share their experiences in IoT projects and compete for prizes. *Hackster* is one of the largest and most active IoT development communities. As of *January* 2021, Hackster has over a million community members, more than $20,000$ of them have at least one published project. Hackster allows the community users to access the different resources about the

[9] `http://help.hackster.io/knowledgebase/hackster-free-store/how-do-i-earn-reputation`

[10] `https://www.hackster.io/store`

IoT community, including platforms,[11] communities,[12] topics,[13] projects,[14] contests,[15] and community users.[16]

2.1 IoT Platforms

A Hackster platform is a group of products that share common hardware and software features. A platform can be a company (e.g., AT&T and Panasonic), a major hardware component (e.g., Arduino and Raspberry Pi), an operating system (e.g., Android), or a cloud backend (e.g., Amazon Web Service). Generic hardware components (e.g., resistors and capacitors) have their own web pages and not considered as platforms. A project may belong to more than one platform. For example, the *J.A.R.V.I.S.: A Virtual Home Assistant*[17] project (shown in Figure 1) belongs to the *Android*, *Arduino*, *Intel*, *Unity*, in addition to three topics, namely *Artificial Intelligence*, *Augmented Reality*, and *Home Automation.*

2.2 IoT projects

Hackster hosts over 26,000 IoT projects. Each project on Hackster maintains a web page containing the information and resources related to that project. Figure 1 shows a snapshot of a sample project hosted on Hackster. We describe the details of projects in the following:

(a) **Project team:** An IoT project can be owned by an individual engineer, a team of engineers, or a company. Each team member of the IoT project has a personal web page (more details in subsection 2.3). Project owners of each project can be displayed without opening the project web page.
(b) **Copyright license:** A different copyright license[18] may be applied to an IoT project. For example, a project may be available under the $Apache-2.0$[19] or $GLP3+$[20] license.
(c) **Description:** Project owners can provide a brief description of the IoT projects. The description gives an overall idea about a project, including its domain and the technology used.
(d) **Project difficulty level:** An IoT project can be classified as *Beginner*, *Intermediate*, *Advanced*, or *Expert*. A more difficult project can be harder to use and replicate.
(e) **Type:** The type of a project indicates the level of detail provided by project owners about an IoT project. Depending on the configuration of the project,[21] a project type can be `Tutorial`, `Protip`, or `Showcase`. The `Tutorial` projects provide step by step instructions, code, and schematics related to the design and implementation of the project. Projects of the `Protip` type show how to solve a single problem with minimal guide. The `Showcase` projects have no or partial instructions but usually contain links to external online resources.
(f) **Estimated Time:** The time required to reproduce the necessary steps to reproduce an IoT project. Such time is estimated by the project owners.
(g) **Views count:** The number of times a project is viewed by the Hackster community users.
(h) **Things:** A set of dedicated lists to show the hardware components, hand tools, software applications, and cloud services that a project uses. Each component may have a link that directs users to an associated web page containing more details about the availability and cost of the component.
(i) **Story:** A project story is composed of sections that provide more details about the purpose of the project, replication steps, and any supplementary materials.
(j) **Schematics:** Project owners may provide a sketch, blueprint, or connection schema that helps users to understand how to assemble the different components together.
(k) **Code:** A project may employ some source code to program, control, or implement a certain functionality of a project. Code may be published as a single file (e.g., `.c` or `.zip`) or uploaded to a remote repository (e.g., GitHub or GitLab).
(l) **Comments:** Users can provide feedback or ask questions to project owners about IoT projects by adding comments to the project web page. Project owners can reply to such comments and may update the web page of the project to address any feedback given by the community users.
(m) **Respects count:** The number of thumbs-up that a project has received from the community users.
(n) **Cover image/video:** Each project on Hackster has a static or animated image that gives a high level picture about the purpose of the project.
(o) **Channels:** The different channels (e.g., communities, platforms, and topics) that a project is connected to. Such channels allow like-minded people to learn and keep up-to-date with the favorite projects and share ideas together.

[11] `https://www.hackster.io/platforms`
[12] `https://www.hackster.io/communities`
[13] `https://www.hackster.io/topics`
[14] `https://www.hackster.io/projects`
[15] `https://www.hackster.io/contests`
[16] `https://www.hackster.io/community`
[17] `https://www.hackster.io/blitzkrieg/j-a-r-v-i-s-a-virtual-home-assistant-d61255`
[18] `https://opensource.org/licenses`
[19] `https://opensource.org/licenses/Apache-2.0`
[20] `https://opensource.org/licenses/gpl-license`
[21] `http://help.hackster.io/knowledgebase/posting-a-project/whats-the-difference-between-a-protip-a-showcase-and-a-tutorial`

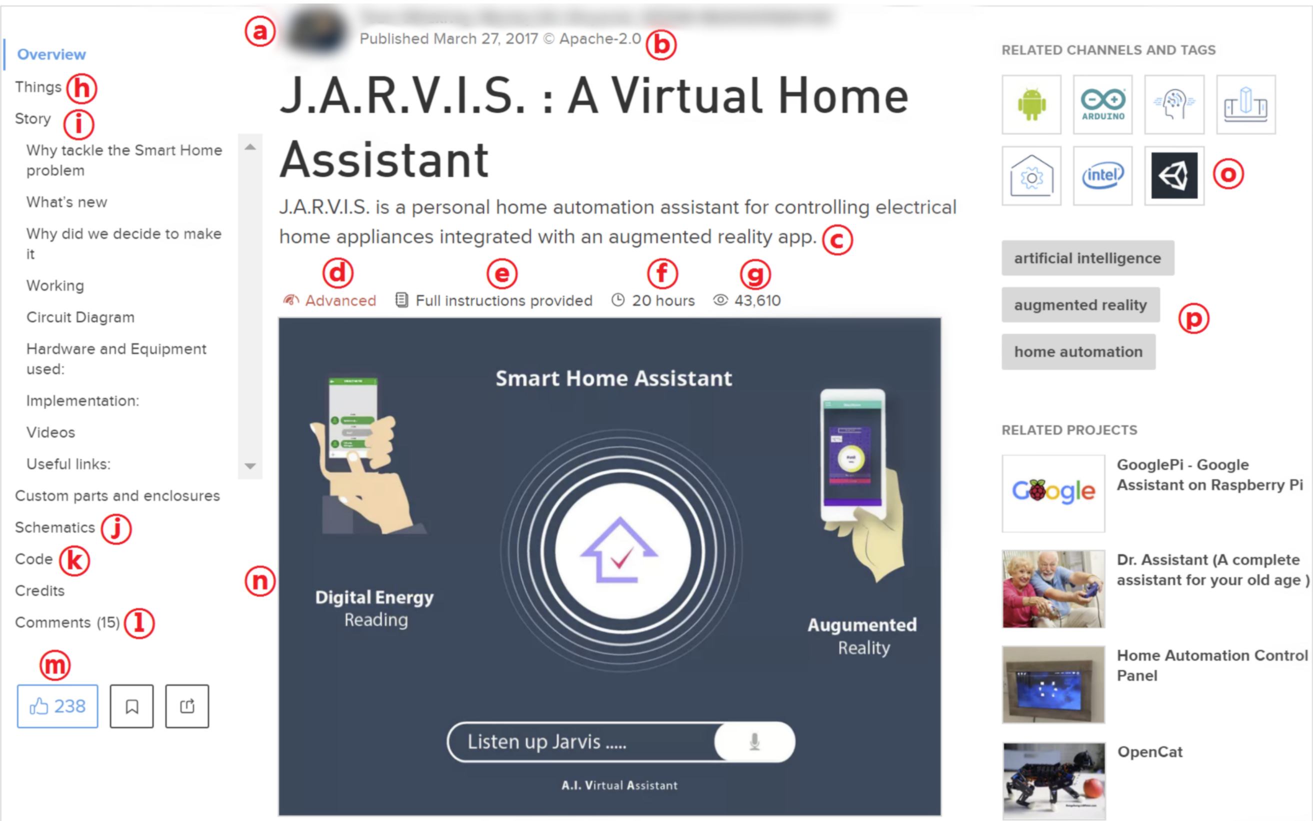


Figure 1: An example of a project web page on Hackster showing the characteristics of IoT projects , including (a) development team (b) copyright license, (c) description, (d) difficulty level, (e) type, (f) estimated replication time, (g) view count, (h) things (i.e., hardware components, hand tools, and services), (i) story (i.e., details about projects), (j) schematics (e.g., blueprints), (k) source code, (l) user comments, (m) respects count, (n) cover image/video, (o) associated channels, and (p) associated tags

(o) **Tags:** Tags are used to describe the purposes, domains, or technologies of a project. Projects can be grouped and accessed by their tags on Hackster.

2.3 IoT community members

Every user on Hackster maintains an own page that shows the personal and professional profile. Each web page contains information about the projects, followers, followings, tools, platforms, awards, and channels of the community users. In addition, users can add short biographies about themselves to show their interests and skills. Hackster maintains a history of all activities[22] performed by the community users.

2.4 IoT Contests

Hackster hosts sponsored contests to allow community members to share their projects for a chance to win prizes. Contests are open for any projects. Each contest maintains an own web page that shows the submission requirements, participating members, submitted projects, prizes, and winners.

2.5 Hackster listing of projects

Hackster allows users to navigate IoT projects of interest by (1) platforms, (2) topics, and (3) products. In addition, users can filter projects based on the project types and difficulty levels. Users can also navigate the featured projects in the community, which are selected by the Hackster team. A project filtration option may lead to hundreds of projects, which are split into pages. At present, Hackster shows only 20 projects per page and users can navigate the rest projects one page after another. Therefore, Hackster enables users to sort the projects, where the projects that are shown in the first page are (i) most recently added, (ii) last updated, (iii) most viewed (referred to as popular), (iv) most respected, or (v) trending. Each page of projects shows a list of project frames, where each frame contains the

[22] `https://www.hackster.io/dixon415/activity`

project cover image/video, title, development team, views count, and respects count. In addition, the project description of each project is shown when hovering over a project frame.

3 Experimental Setup

This section presents the experimental setup of our empirical study. We explain how we collect and prepare the data for our studied RQs.

3.1 Data Collection

Figure 2 gives an overview of our study, which is based on data collected from Hackster. We develop a crawler that collects data related to 19,083 IoT projects hosted on Hackster. Out of these projects, we select the projects that are active on *January 31, 2020*. Additionally, we exclude (a) the older versions of projects in the case of duplicates, (b) projects that do not have project owners, and (c) the projects of which project owners are no longer members of Hackster. As a result, our dataset contains 18,299 IoT projects from 206 platforms.

Hackster does not preserve historical data about how projects gain popularity over time. Therefore, we run our crawler once every day to monitor the changes to the popularity measures of the studied IoT projects. We kept the crawler running for six months (from *August 1, 2019* to *January 31, 2020*). In this particular analysis, we only include the projects that were published before *August 31, 2019* and remained active until *January 31, 2020*. We use this selection criterion to make sure that all projects have the required data points (i.e., daily views, respects, and trending scores) for the entire period. In total, we obtain the daily views count, respects count, and trending scores of the selected projects for 183 days.

In addition, we collect meta-information about the projects, including descriptive characteristics, hardware information, project owners, and user feedback. Considering that an IoT project can be associated with multiple platforms (e.g., hardware or service providers), we create an independent record for a project and each particular platform associated with the project. For example, a project that is associated with 4 platforms has 4 records in our dataset, each with a different platform. In addition, we collect the number of projects and the number of members of each platform and assign them to each replicated record in the dataset. As a result, our dataset contains 26,596 records.

3.2 Data Processing

In this subsection, we explain how we process the data of the selected IoT projects. First, we show how we compute the three popularity measures (i.e., the dependent variables) of the projects. Next, we discuss the factors computed in our study to model our dependent variables. For each computed factor, we aggregate it to the project level when necessary.

3.2.1 Computing popularity measures

Hackster ranks the hosted IoT projects according to (a) the project *views*,[23] (b) the project *respects*,[24] and (c) the project *trending scores*.[25] Hackster uses the term *popular* to sort IoT projects according to the number of views. However, prior studies have considered various measures to assess the popularity of GitHub projects and mobile applications, such as the number of *downloads* [3], *stars* [9], *watchers* [49], *forks* [70], and *ratings* [39]. As reported by previous studies, a higher number of views may not always indicate that a project is well-perceived by the community. For example, the Arduino Thermometer[26] project is one of the most highly viewed IoT projects on Hackster (with over 378,000 views). However, the Arduino Thermometer project has received just above 120 respects (i.e., 0.3% of the views count). Conversely, although the Christmas Gift Box[27] project has 4,200 views, it is highly ranked in terms of the number of respects, with over 600 respects (i.e., 14% of the views count).

We perform a Pearson's correlation test [31] between the popularity measures, to verify to what extent such measures are correlated. We observe that the number of views and respects are not highly correlated (i.e., Pearson's coefficient of 0.38). In particular, the median views count of the studied projects is 1,000, whereas the median respects count is six. Such a gap may indicate that out of 1,000 users viewing the same project, only six of them will give the project a thumbs-up. In addition, we observe that the views count and respects count have relatively lower correlations with the project *trending score* (i.e., Pearson's coefficients of 0.23 and 0.31). Moreover, a higher number of views may indicate that a project has reached more users, which could be due to the use of advertisements. While making a project more reachable is important, project owners still desire to gain a positive user perception. Therefore, it is important to understand the factors that have relationships with each of the three popularity measures of IoT projects. To this end, we rank the IoT projects by sorting them in a descending order using each of the popularity measures.

[23] https://www.hackster.io/projects?sort=popular
[24] https://www.hackster.io/projects?sort=respected
[25] https://www.hackster.io/projects?sort=trending
[26] https://www.hackster.io/TheGadgetBoy/ds18b20-digital-temperature-sensor-and-arduino-9cc806
[27] https://www.hackster.io/31000/christmas-gift-box-0ff17e

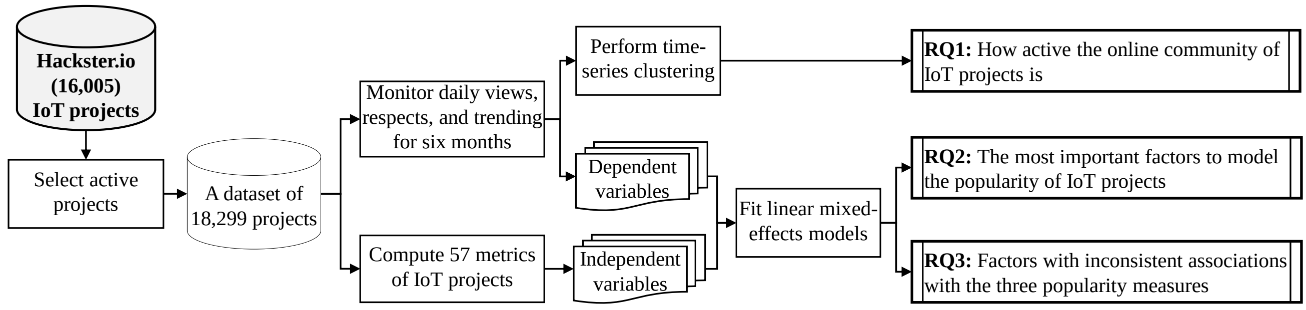


Figure 2: Overview of our study

3.2.2 Computing Independent Variables

This step is concerned with the selection of the factors that are used as independent variables in our models. We use those variables to model and explain our dependent variables. In Table 1, we present four groups of factors for which we study the relationship with the popularity of IoT projects: Project Description (i.e., 21 factors), code & Hardware (i.e., 12 factors), Project owner(s) (15 factors), and Comments (i.e., 9 factors). Hackster provides guidelines[28] to help project owners publish their projects. The guidelines contain directions on how to use seven factors, namely cover image, difficulty, tags, team, hardware, story, schematics, and code. We use all the aforementioned factors as independent variables in our models. We describe each factor in the last column of Table 1. We explain below how we compute the factors in the following:

Project Description factors: We compute 21 factors that we extract from the project web pages on Hackster. Out of these factors, only the *name*, *cover image/video*, *brief description*, and *project owners* are shown on the web pages that list the projects.[29] More details about projects (e.g., type, channels, tags, and license) can be shown inside the specific project web page. In addition, Hackster curates a set of projects and marks them as featured projects. These projects can be browsed by users using a dedicated list on Hackster. We identify the projects that are selected as featured projects and investigate whether the popularity of IoT projects is associated with the decision of being featured projects. Moreover, we identify the projects that participate in the Hackster online contests and have won any of the contests. Project owners may have control over the majority of the factors under the Project Description group, and therefore the popularity of IoT projects can be improved by considering such factors. Hackster provides guidelines to project owners on how to use these factors properly. Providing precise, descriptive, and attractive project details can help projects to gain more popularity.

Hardware factors: We compute 12 factors that we extract from the *Things* section[30] of the project web page. These factors represent measurements related to the technical details of the project, including the required software, hardware, and hand tools. It is important for project owners to publish every particular detail of the hardware used in their projects so that users can reproduce the project. We count the number of hardware components required to reproduce the project. We also count the total quantity of hardware components. For example, a project may require 3 *sensors* and 2 *cameras*. Hence, the number of hardware components is 2 (i.e., a sensor and a camera) and the number is 5 pieces. Considering that hardware components can be purchased from different online hardware suppliers (e.g., Amazon or eBay), we compute the number of unique hardware suppliers and the most common suppliers within every project. For every supplier, we extract only the domain name of the website (i.e., we eliminate the country-based versions and sub-pages of the website). For example, from '`https://www.amazon.ca/camera`', we extract only '`amazon`' as an online hardware supplier. Project owners can control the factors of this group by listing every particular hardware component required to complete the project. In addition, project owners can provide users with the best alternatives to purchase the hardware. Using proper hardware components (in terms of reliability, cost, and purchase alternatives) can be associated with project popularity.

Project owner(s) factors: We compute 15 factors that we extract from the web pages of the project

[28] `https://www.hackster.io/guidelines`
[29] `https://www.hackster.io/projects/?page=1`
[30] `https://www.hackster.io/gbarbarov/open-led-race-a0331a#things`

Table 1: Dimensions of factors used as independent variables in our linear mixed-effects models

	Factor	**Data type**	**Description**
Project Description factors	Platform	Factor	The platform (e.g., hardware or cloud services) that the project belongs to
	Number of platform projects	Factor	The number of projects that belong to the platform of each project
	Number of platform members	Factor	The number of project owners who are members of the platform of each project
	Listed in the featured projects	Numeric	The page number where the project is listed in the featured projects, if any.
	Project type	Factor	The type of the project (e.g., TUTORIAL, PROTIP, WIP, or SHOWCASE)
	Project cover	Factor	The main cover (i.e., image or video) used to demonstrate the project
	Project age	Numeric	The number of days since the project was published online
	Last modified	Numeric	The number of days since the project was modified
	Difficulty level	Factor	The difficulty level of the project (e.g. Easy, Intermediate, Advanced, or Expert)
	Copyright license	Factor	The license to distribute a project (e.g. GPL3+, Apache2.0, MIT, CDDL1.0, etc.)
	Number of project owners	Numeric	The number of team members who developed the project
	Number of tags	Numeric	The number of tags (e.g., robotics, automation, or security) associated to a project
	Number of channels	Numeric	The number of channels (e.g., communities and topics) a project is connected to
	Number of contests	Numeric	The number of contests the project submitted to
	Won a contest	Factor	Whether the project has won one or more contests
	Description length	Numeric	The number of characters used to describe the project
	Number of videos	Numeric	The number of (YouTube) videos used in the project page
	Number of images	Numeric	The number of images used in the project page
	Number of story sections	Numeric	The number of sections used to explain how the project works
	Length of story	Numeric	The number of words used to explain the project story
	Number of links in story	Numeric	The number of links used in the project story
	Number of schematics	Numeric	The number of documents used to show the sketches/blueprints of the project
Hardware factors	Hardware components	Numeric	The number of (unique) hardware components the project is composed of
	Quantity of hardware components	Numeric	The total quantity of hardware components needed to complete the project
	Hand & fabrication tools	Numeric	The number of hand tools required to fabricate the project
	Number of purchase links	Numeric	The number of links used to purchase the hardware components and tools
	Number of unique hardware suppliers	Numeric	The number of unique suppliers of hardware components
	Most common hardware supplier	Factor	The most common supplier (e.g., Amazon or eBay) to purchase hardware
	Vendors per hardware components	Numeric	The maximum number of vendors to purchase a particular hardware component
	Estimated time	Numeric	The time (in seconds) required to reproduce all the necessary steps of the project
	Estimated cost mentioned	Factor	Whether the hardware cost is indicated in the project description or story
	Tools without links to purchase	Numeric	The number of tools that do not have purchase links
	Hardware-controlling code	Factor	Whether the project code is available as a file or on a repository (e.g., GitHub)
	Software applications & services	Numeric	The number of software programs required by the project
Project owner(s) factors	Length of project owner's biography	Numeric	The average (mean) number of words in project owner biography(ies)
	Personal Web page available	Factor	Whether the project owner(s) of the project have a link to a personal Web page
	Geographical location	Factor	The country where the project owner(s) of the project reside
	Project owner's projects	Numeric	The unique number of projects published by the project owner(s) of the project
	Project owner's followers	Numeric	The unique number of users who follow the project owner(s) of the project
	Project owner's followings	Numeric	The unique number of users followed by the project owner(s) of the project
	Project owner's skills	Numeric	The unique number of skills of the Project owner(s) of the project
	Project owner's tools	Numeric	The unique number of tools used by the project owner(s) of the project
	Project owner's channels	Numeric	The unique number of channels the project owner(s) are connected to
	Project owner's communities	Numeric	The unique number of communities the project owner(s) are members of
	Project owner's contests	Numeric	The unique number of contests the project owner(s) participated in
	Project owner's awards	Numeric	The unique number of awards the project owner(s) obtained
	Project owner's respects	Numeric	The total number of respects the project owner(s) give to other projects
	Project owner's comments	Numeric	The total number of comments by the project owner(s) to other projects or posts
	Project owner's likes	Numeric	The total number of likes the project owner(s) give to other users' posts
Feedback factors	Number of comments	Numeric	The number of user comments raised in the project page
	Number of replies	Numeric	The number of replies to user comments
	Ratio of replies to comments	Numeric	The ratio of replies to user comments
	Number of project owner's replies	Numeric	The number of replies provided by the project's project owner(s)
	Ratio of project owner's replies to comments	Numeric	The ratio of the project owner's replies to user comments raised
	Ratio of project owner's replies to replies	Numeric	The ratio of the project owner's replies to user replies
	Number of positive comments	Numeric	The number of comments that contain positive feedback from users
	Number of neutral comments	Numeric	The number of comments that contain neutral feedback from users
	Number of negative comments	Numeric	The number of comments that contain negative feedback from users

owners.[31] Project owner(s) factors are related to the technical and social details of project owners. This group includes the activities and interactions of project owners with the other parties of the Hackster community. For those projects that have multiple owners, we aggregate the computed values by considering all project owners. For example, we take the unique number of followers and skills of all project owners. Project owners have the ability to change or update the information related to the project owner(s) factors. Maintaining well-descriptive information about project owners and building more connections with the community may have an association with the popularity of a project.

Feedback factors: We compute 9 factors that we extract from the *Comments* section[32] of the web pages of projects. *Comments* allow users to discuss with the project owners about any concerns or questions related to the published project. The number of comments depends on other users to post comments. Project owners may reply to the posted comments and address any raised concerns. We distinguish user comments from owner replies to investigate which metrics are more associated with the popularity of a project. Comments posted on a project web page can be positive (e.g., praise), negative (e.g., critique), or neutral (e.g., a question). To distinguish between the different sentimental types of comments, we apply sentiment analyses [56] using the SentiStrength-SE tool [26]. We compute a sentiment score for each comment to identify whether it is positive, negative, or neutral. The SentiStrength-SE tool assigns a score to each comment that ranges from -5 (the lowest) to $+5$ (the highest). According to the SentiStrength-SE tool, (a) sentiment scores of -1 or $+1$ indicate a neutral comment, (b) sentiment scores of $\{+2, +3, +4, +5\}$ indicate a positive comment, and (c) sentiment scores of $\{-5, -4, -3, -2\}$ indicate a negative comment. We use the generated scores of all comments of a project to compute the number of positive, negative, and neutral comments. Receiving constructive feedback from community users and maintaining consistent responses to user comments may help a project to be popular.

3.2.3 Correlation and Redundancy Analysis

Regression models can be adversely affected by the existence of highly correlated and redundant independent variables [17]. Therefore, we perform correlation and redundancy analyses for the independent variables used in our models. We follow the guidelines that are provided by Harrell [24] to train regression models.

Correlation Analysis: In this step, we employ the Spearman rank ρ hierarchical clustering analysis [47] to remove highly correlated variables in each of the subject projects. Hierarchical clustering is a pairwise analysis and helps detect variables that have positive or negative relationships and thus a single variable can be sufficient to represent another variable To this end, we use the `varclus` function from the `rms`[33] *R* package. For each pair of independent variables within all clusters that have a correlation of $|\rho| > 0.7$, we remove one variable and keep the other variable in the models. According to the principle of parsimony in regression modeling, simple explanatory variables should be preferred over complex variables [61]. Given that our explanatory variables are equally simple (e.g., in terms of computation), we keep the variables that are more informative [22]. In our case, we keep variables that convey more information about the IoT projects or the project owners. For example, the *Hardware items needed* factor is highly correlated with *Quantity of hardware components*, the *Number of purchase links*, and *Number of unique hardware suppliers*. Therefore, we keep the *Hardware items needed* variable, since it is more descriptive than the other three variables. Similarly, the *Project owner's tools* is highly correlated with the *Project owner's channels*. Therefore, we keep the *Project owner's channels*, since the channels that a project owner belongs to can better describe the connections of a project owner than the tools that a project owner uses.

In Figure 3, we show the dendrogram of the hierarchical clustering of independent variables for the subject projects. In this dendrogram, we observe six clusters of highly correlated variables ($|\rho| > 0.7$). We distinguish each cluster of highly correlated variables with a different color. In Table 2, we present the highly correlated variables and the variable we select in each cluster. After removing the *Developer's projects* variable, we find that the *Developer's followers* variable becomes highly correlated with the *Developer's comments* variable (i.e., an additional cluster). For this resulting cluster, we remove the *Developer's comments* variable and keep the *Developer's followers* variable.

Redundancy Analysis: In this step, we perform a redundancy analysis on the remaining 44 independent variables (i.e., those that survive the correlation analysis step). Redundant variables can distort the relation-

[31] `https://www.hackster.io/anthony-ngu`

[32] `https://www.hackster.io/saifalikabi/digital-logic-board-03fd26#comments`

[33] `https://cran.r-project.org/web/packages/rms/rms.pdf`

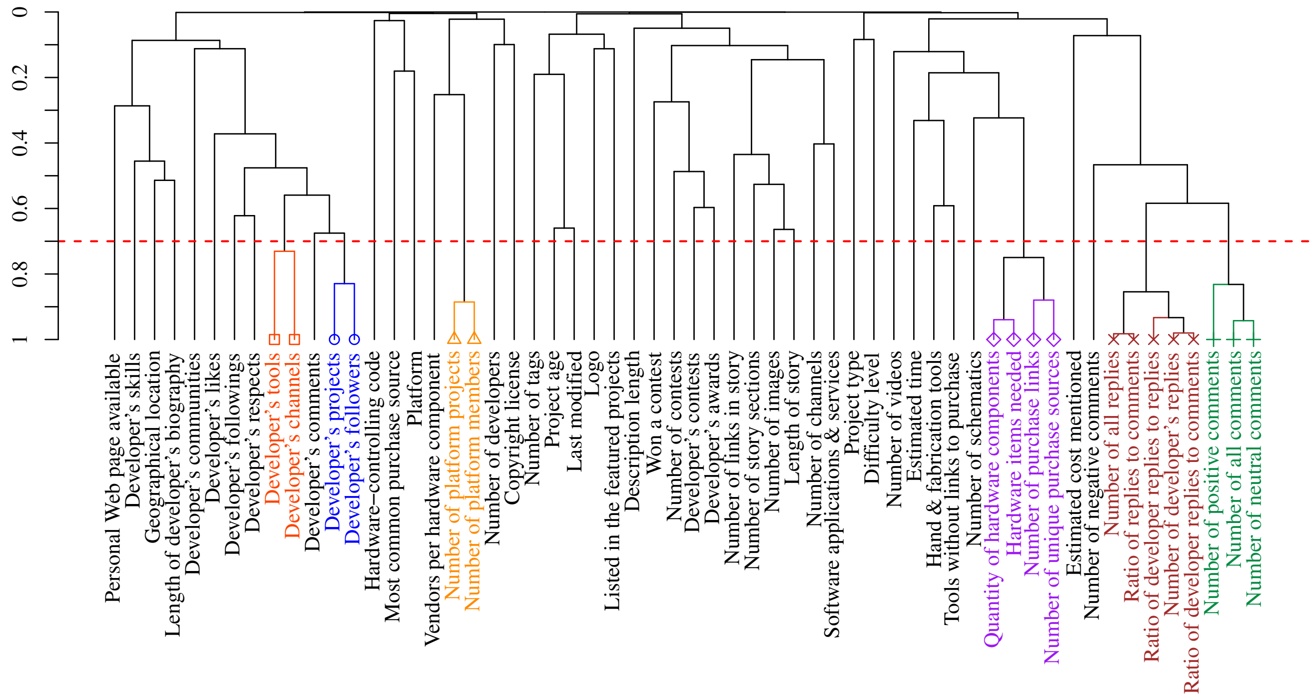


Figure 3: The hierarchical clustering of independent variables in the studied projects. We use color codes to distinguish variables of highly correlated clusters from each other (i.e., colors have no special meaning).

Table 2: Selected variables of the highly correlated variables in the projects

	Cluster of highly correlated variables	**Selected variable**
1	Project owner's tools Project owner's channels	Project owner's channels
2	Project owner's projects Project owner's followers	Project owner's followers
3	Number of platform projects Number of platform members	Number of platform projects
4	Hardware components Quantity of hardware components Number of purchase links Number of unique hardware suppliers	Hardware components
5	Number of all replies Number of project owner replies Ratio of replies to comments Ratio of project owner replies to comments Ratio of project owner replies to replies	Number of project owner replies
6	Number of all comments Number of positive comments Number of neutral comments	Number of positive comments

ship between the dependent variable and the other independent variables [24]. To this end, we use the `redun` function from the `rms` *R* package, which performs a parametric additive model to detect the variables that can be predicted from all the other variables. If an independent variable can be predicted by other independent variables with an $R^2 \geq 0.9$, we discard such a variable [24]. Our redundancy analysis reveals no further redundant variables in our dataset.

4 Experimental Results

In this section, we discuss the motivation, the approach, and the findings of our research questions.

4.1 ***RQ₁*: *How active is the online community of IoT projects?***

Motivation. Online IoT communities, such as Hackster, host thousands of hardware projects published from a wide range of technologies. Such projects are listed in a series of ranked web pages. Projects that are displayed at the beginning of the list are more likely to be selected by the community users to browse. Therefore, understanding the evolution of the popularity of projects helps project owners to understand the odds of rising and falling in popularity over time. It is also important to explore whether project views and respects maintain a different evolution. As such, project owners can take a wiser decision when publishing new projects or trying to gain popularity for currently published projects. In this RQ, we investigate the evolution patterns of the three popularity measures of IoT projects (i.e., views count, respects count, and trending scores).

Approach. To gain a better understanding of IoT projects in our dataset, we analyze the popularity of the studied IoT projects (a) over time, (b) per platform, and (c) per tag. We perform the following analyses:

- We use time series clustering [35] to infer the evolution of the three popularity measures of IoT projects. We use a moving average to smooth the fluctuating values of daily views, respects, and trending scores. In our time series clustering, we use the following:

  Optimum number of clusters. We use the gap statistic approach [58] to estimate the optimum number of clusters. The gap statistic uses the output of a clustering algorithm (e.g., k-means [36]) and compares it with the change in a within-cluster dispersion. The procedure tries different numbers of clusters to maximize the gap statistic value. We apply the gap statistics algorithm using the `clusGap` function in the `cluster`[34] R package. We use a range of numbers of clusters, k, between 2 and 50. Then, we select the smallest k number of clusters at which the rate of increase of the gap statistic begins to decline. As a result, we obtain the following optimum numbers of clusters: ***five*** clusters of daily views, ***seven*** clusters of daily respects, and ***three*** clusters of daily trending scores.

  Distance function. We use the Dynamic Time Warping (DTW) method [6] to measure the similarity (i.e., distance) between two time-series vectors of a certain popularity measure. DTW aligns two time series (e.g., daily views of IoT projects) in a way that the differences between the two time series are minimized. Equation 1 shows the distance of the warping path between two time series:

  $$dist_{DTW} = \frac{\sum_{i=1}^{\kappa} \omega_i}{\kappa} \tag{1}$$

  DTW builds a warping path $W = \{\omega_1, \omega_2, ..., \omega_\kappa\}$ where κ denotes the number of points in W and $max(m, n) \leq \kappa \leq m + n - 1$ [35].

  Partitional clustering. We use the Partitioning Around Medoids (PAM) (also known as *k-medoids*) to partition time-series popularity measures into n clusters, where each cluster contains at least one object and each object belongs to one cluster. PAM chooses data points as centers (i.e., *medoids*) with arbitrary distances. We use DTW as the distance function and the optimum number of clusters identified using the gap statistic.
- We analyze projects that are published on the same day to investigate whether they have gained similar popularity after a while. We use boxplots [65] to show (a) the distribution of projects published daily on Hackster and (b) the popularity distributions of projects published on the same day. Boxplots are visual representations of the minimum, lower quantile, median, upper quantile, and maximum of the members in each group of values.
- We use the WordCloud[35] R package to visualize the frequency of all projects that use the platforms and tags on Hackster. The larger the size of a platform or tag in the cloud, the more the frequency of projects that use such a platform or tag. In addition, we identify the types of projects (e.g., health, gaming, or tracking) in each of the generated clusters of projects. Tags assigned to each project may not

[34] https://cran.r-project.org/web/packages/cluster/cluster.pdf
[35] https://cran.r-project.org/web/packages/wordcloud/wordcloud.pdf

only refer to the type of such a project, but also the technology or tool employed by that project (e.g., *bluetooth*, *sms*, and *data collection*). Hence, when visualizing word clouds and profiling clusters, we eliminate the tags that are too generic and do not reflect the type of project.

Findings. ***We observe an increasing trend of publishing IoT projects on Hackster with a median of* 11 *projects published everyday.*** Figure 4 shows boxplots that represent, for each month, the distribution of the number of projects published everyday during the period between *May 2013* and *February 2020*. For example, by looking at the projects that are published on *August 21, 2018*, we observe that there were 147 projects published, which is the maximum number of project publications throughout the entire period. In Figure 5, we show box plots representing the distributions of the views count, respects count, and trends of the projects published on *August 21, 2018*. We observe that the projects do not gain similar popularity throughout the three-month period. Such results research to investigate the factors that may have an association with the popularity of IoT projects.

IoT projects are viewed by the community users in five different patterns. Figure 6 shows the centroids of the five identified clusters of daily views of IoT projects. In Table 3, we present the profiles of the daily views clusters (extracted from the word clouds attached in the Appendix), in addition to the frequency (i.e., the number and percentage of projects) of each cluster. We observe that the majority (i.e., 41%) of projects gain a reasonably rising number of views during the early period (e.g., the first month) of publication (see Cluster #5 in which projects target health, nutrition, and occasions, such as *halloween* and *christmas*). However, the number of views starts to decline for the next few months but rises again at later stages. There are 16% of the projects that maintain a linearly increasing number of views over time (see Clusters #1 & #2 in which projects use computer vision techniques to track the health of plants). However, about two-thirds of such projects may start losing views at later stages. Finally, the views of 28% of the projects follow a declining pattern over time (see Cluster #3 in which projects target kids' health and gifts). Although the types of projects in Cluster #3 are somewhat similar to those projects of Cluster #5, we observe a difference in the platforms of such projects, i.e., the majority of projects of Cluster #5 use Arduino and RaspberryPi boards, whereas the majority of projects of Cluster #5 use RaspberryPi and Particle. Our findings indicate that updating the information of projects after publication can help projects gain a steady number of views.

Two-thirds of the projects receive no respects from the IoT community users. Figure 7 shows the centroids of the seven identified clusters of daily respects of IoT projects. In Table 3, we present the profiles of the daily respects clusters (extracted from the word clouds attached in the Appendix), in addition to the frequency (i.e., the number and percentage of projects) of each cluster. We observe that the majority (i.e., 66% percent) of the projects have received zero respects from the community users (See Cluster #1 in which projects mainly focus on tracking the health of kids and plants). Such a result may indicate that viewing a project may not necessarily indicate that the project is well perceived by the community users. In addition, we observe that projects may experience an early spike of respects (Cluster #2 with 18% of the projects that provide solutions for kids entertainment and health tracking) or a late spike of respects (Cluster #7 with 5% of the projects that use drones to monitor crops). There are almost no projects that have received a steady rise of respects. Projects that fall under Cluster #5, representing 6% of the projects that have a potential of receiving a late-rising number of respects. Those projects cover different types of activities, including entertainment, health, and agriculture. Still, rising of such projects is subject to a sudden decline afterwards.

Projects are mainly top-trending at the early stages of their publication. Figure 8 shows the centroids of the 3 identified clusters of daily trending pages of the studied projects. In Table 3, we present the profiles of the trending clusters (extracted from the word clouds attached in the Appendix), in addition to the frequency (i.e., the number and percentage of projects) of each cluster. As Figure 8 depicts, almost all projects do not maintain a high trending score all the time. However, the trending scores do not fall at the same rate. The majority (i.e., Cluster #3 with 58% of the projects) of kids health and gifts projects maintain a linear decrease in the trending score, which leads projects to be listed at later pages among all Hackster projects. 42% of the projects (i.e., Cluster #1 and #2, which compose of projects that mainly track the health of plants) experience an early (i.e., in the first 25-40 days of publication) fast decrease in the trending score. Yet, half of those projects maintain a steady trending score for the rest of the day, whereas trending scores of the other half of projects continue to decrease at a slow pace.

Featured projects are updated frequently on Hackster (almost daily). Figure 9 shows the number of featured projects over time. Hackster features certain IoT projects on dedicated web pages. The selection of

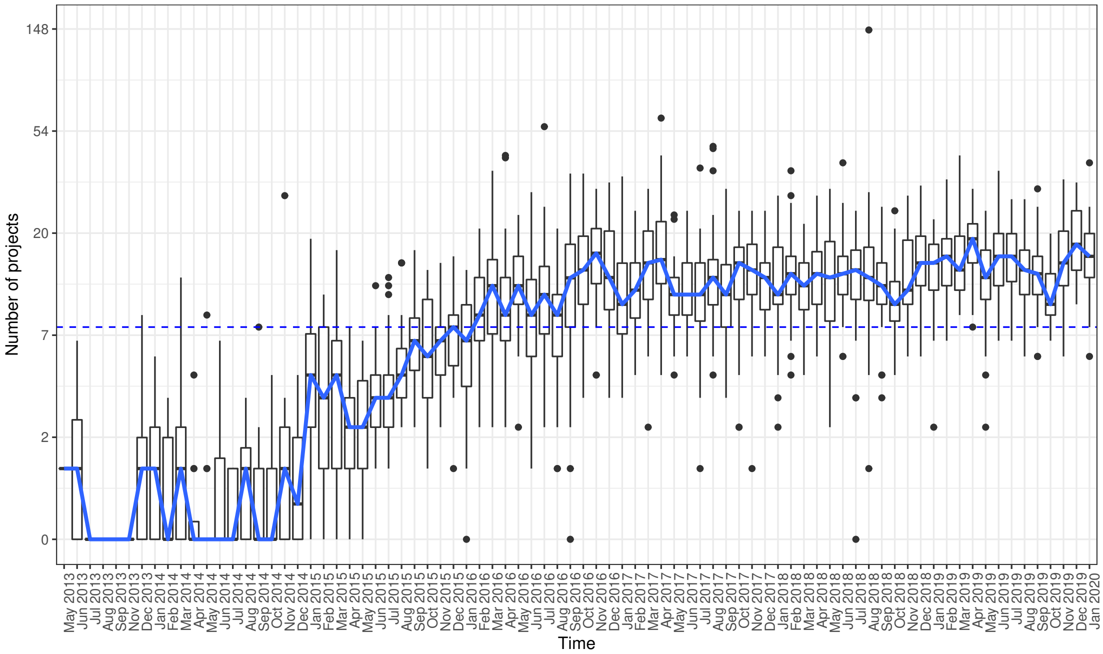


Figure 4: The distribution of projects published daily between May 2013 and February 2020

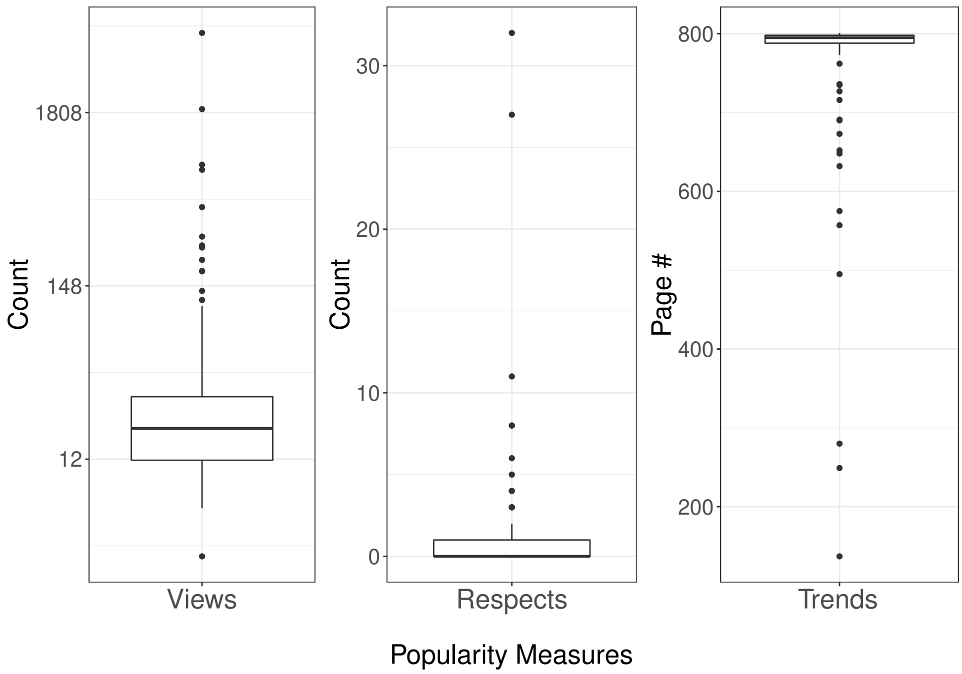


Figure 5: The distributions of the views count, respects count, and trending pages of the projects published on August 21, 2018

such projects is conducted by the Hackster community organizers, but the selection criteria are not disclosed. We observe that the pages of featured projects are always increasing by a factor of $0-6$ projects every day. In addition, we observe that the featured projects remain featured and never reconsidered for removal from the featured set of projects.

The majority of IoT projects are associated with the Arduino, RaspberryPi, SparkFun, and Adafruit platofrms. Figure 10a shows a word cloud that demonstrates the frequency of projects that use the 206 platforms in our dataset. Over 25% of the projects use the Arduino platform, followed by RaspberryPi, which is used by about 15% of the projects. In addition, we observe that 17% of the projects that use Arduino also use

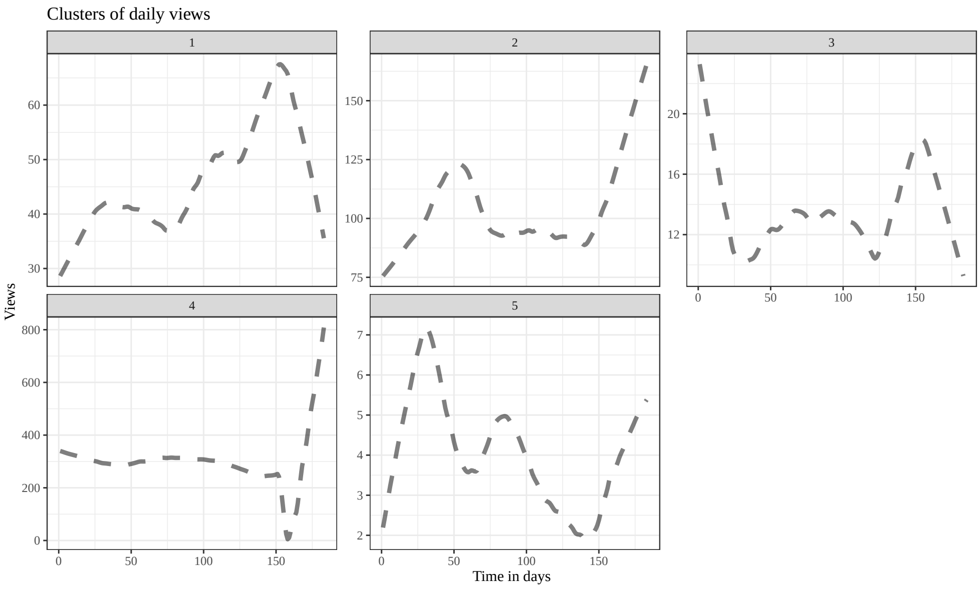


Figure 6: Centroids of the clusters of daily views. X-axis denotes the days and Y-axis shows the number of views

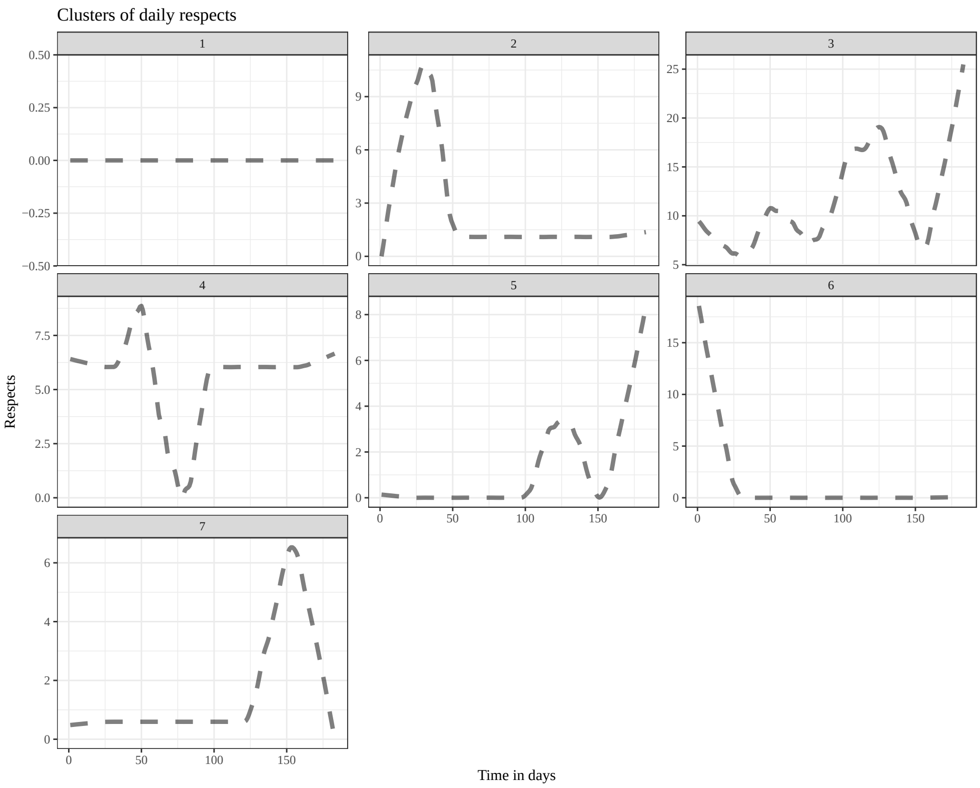


Figure 7: Centroids of the clusters of daily respects. X-axis denotes the days and Y-axis shows the number of respects

SparkFun or Adafruit or both of them. Similarly, 19% of the projects that use RaspberryPi also use Microsoft, SparkFun, Adafruit or a combination of them. Furthermore, we observe that the projects in our dataset use a median of 3 platforms.

The `robotics` and `led` tags are the most frequently assigned to IoT projects. Figure 10b shows a word cloud that demonstrates the frequency of tags assigned to the projects in our dataset. Our dataset contains $2,001$ tags that are related to IoT technology. We show in Figure 10b 200 tags with the most number of projects that use them. The number of projects that use the robotics and led tags is $1,113$ and $1,053$, respectively. We observe that the projects in our dataset use

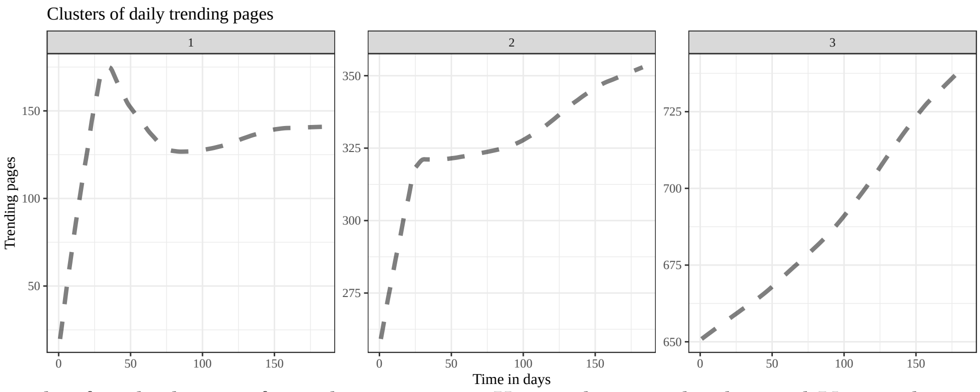


Figure 8: Centroids of each cluster of trending patterns. X-axis denotes the day and Y-axis shows trending scores

Table 3: Summary of the clusters of daily views, respects, and trending scores. The types of projects are summarized using the platforms and tags word clouds in the Appendix.

	Cluster	Pattern	#	%	Profile
Views	1	Rising - Late Falling	2,929	16%	Arduino/SparkFun : health, tracking, kids, drones, garden
	2	Rising	1,777	10%	Arduino/SparkFun : tracking, health, solar, computer vision, healthcare
	3	Falling	5,082	28%	Arduino/RaspberryPi : health, garden, kids, halloween, machine learning
	4	Maintaining - Late Rising	942	5%	Arduino/SparkFun : computer vision, home security, tracking, garden, health
	5	Early Rising - Falling - Later Rising	7,569	**41%**	RaspberryPi/Particle : health, food and drinks, kids, halloween, christmas
Respects	1	Zero-Maintaining	12,091	**66%**	RaspberryPi/Arduino : health, kids, food and drinks, garden, tracking
	2	Early Spike	3,368	18%	Arduino/SparkFun : health, tracking, computer vision, kids, halloween
	3	Rising	198	1%	Arduino/SparkFun : home security, machine learning, drones, solar, pets
	4	Sudden Collapse	79	≈0%	Arduino/SparkFun : tracking, transportation, kids, drones, energy efficiency
	5	Late Rising	1,018	6%	Arduino/SparkFun : machine learning, drones, garden, entertainment, healthcare
	6	Early Falling	689	4%	Arduino/RaspberryPi : garden, computer vision, energy efficiency, real time, health
	7	Late Spike	856	5%	Arduino/SparkFun : health, drones, tracking, computer vision, irrigation
Trending	1	Early Fast Falling - Recovering	4,064	22%	Arduino/RaspberryPi : tracking, health, drones, solar, garden
	2	Early Fast Falling - Slow Falling	3,712	20%	Arduino/SparkFun : health, tracking, garden, kids, machine learning
	3	Fast Falling	10,523	**58%**	RaspberryPi/SparkFun : health, kids, halloween, garden, christmas

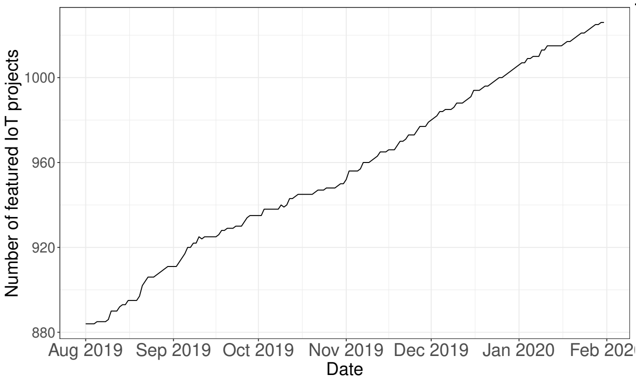


Figure 9: The number of featured projects over time

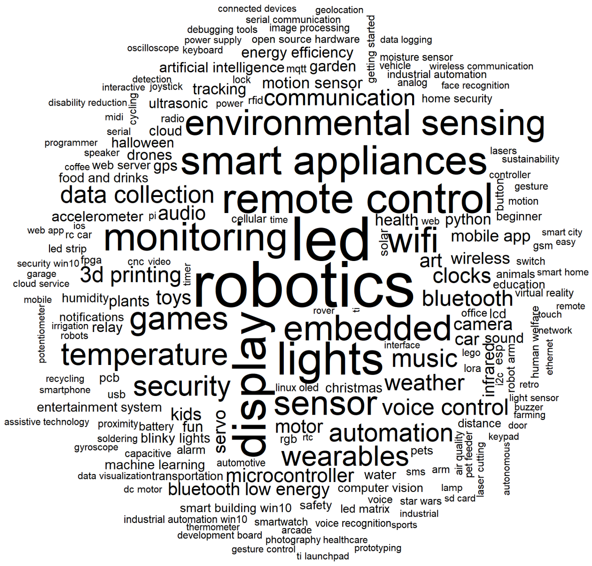
a median of 2 tags. This result indicates that the majority of projects strictly follow the guidelines provided by Hackster, which encourages projects owners to use a maximum of three tags. However, over 3,000 (i.e., 20%) of the projects still assign more than three tags.

IoT projects are increasingly published every day. The view and respect rates of IoT projects can vary dramatically over time. The Arduino and RaspberryPi hardware platforms are commonly used by a large number of IoT projects. There are 20% of IoT projects that do not follow the community guidelines (e.g., assign more tags than required, i.e., three).

4.2 ***RQ$_2$*: *What are the most important factors to model the popularity of IoT projects?***

Motivation. Studying the factors that are associated with the popularity of IoT projects is important because it helps (new) IoT engineers to better work on an IoT project before or after publication. By understanding the factors that have an association with the popularity of IoT projects, project owners can attract more users and foster new business opportunities. Therefore, in this RQ, we aim to understand the important factors to model the popularity of IoT projects while controlling the age and platforms of projects.

(a) Frequency of IoT projects per each platform



(b) Frequency of IoT projects in each tag

Figure 10: Word clouds of IoT platforms and tags on Hackster

Approach. To control the potential variation of the platforms and publication dates of the studied projects, we fit linear mixed-effects regression models to study the three popularity measures of IoT projects. We build three separate linear mixed-effects models to understand the factors that are strongly associated with each popularity measure of the project. Mixed-effects models allow assigning (and estimating) a different intercept for each project [63] to control the variance between projects in terms of age and platforms. Considering that we aim to study the relationships between the popularity of IoT projects and the factors listed in Table 1, we particularly use the generalized mixed-effects models for linear regression. Generalized mixed-effects models are statistical regression models that contain both fixed and random effects [19]. Fixed effects are variables with constant coefficients and intercepts for every individual observation. Random effects are variables that are used to control the variances between observations across different groups (i.e., project platforms and ages). Our linear mixed-effects models assume a different intercept for each group [32]. Traditional regression models, in contrast, use fixed effects only, which disregards the variances of the popularity of IoT projects across different platforms and ages.

Equation 2 shows the equation of the linear mixed-effects model. In Eq. 2, Y_g denotes a project popularity measure; β_0 demonstrates the constant intercept; X_i represents the independent variables; β_i represents the coefficients of each X_i; ϵ_g indicates the errors; and θ_g represents the intercepts that vary across each platform and age. We use the `lmer` function in the `lme4` `R` package to use linear mixed-effects models.

$$Y_g = \beta_0 + \theta_g + \sum_{i=1}^{n} \beta_i X_i + \epsilon_g \tag{2}$$

Significant independent variables are marked with asterisks in the output of the mixed-effects models using the `ANOVA` test [44]. An independent variable is significant if it has $Pr(<|\chi^2|) < 0.05$. $Pr(<|\chi^2|)$ is the *p-value* that is associated with the χ^2-statistical test. The χ^2 (Chi-Squared) values show whether a model is statistically different from the same model in the absence of a given independent variable according to the degrees of freedom in the model. The higher the χ^2, the higher the explanatory power of an independent variable. We use *upward* ($\nearrow$) and *downward* ($\searrow$) arrows to indicate whether an independent variable has a direct or inverse relationship, respectively, with the dependent variables (i.e., the project popularity measures).

We compute the number of Events Per Variable (EPV) or our dataset to assess the risk of overfitting our models [42]. EPV values represent the ratio of the number of records in our dataset to the degrees of freedom (i.e., the number of independent variables plus the number of levels in the categorical variables). A dataset with an EPV above 10 is less likely to run into an overfitting problem [42].

We evaluate the performance of the models using the marginal and conditional R^2 values:

- The *marginal* R^2 is a measure of the goodness-of-fit of our mixed-effects models. It represents the proportion of the total variance explained by the fixed effects [37]. Higher values of the *marginal* R^2 indicate that fixed effects can properly explain the dependent variable.

- The *conditional* R^2 is another goodness-of-fit measure of the linear mixed-effects models. It represents the proportion of the variance explained by both fixed and random effects [37]. Higher values of the *conditional* R^2 indicate that the proportion of the variance that is explained by both fixed and random effects is higher than the proportion of the variance that is explained by fixed effects only. A high difference between the values of *conditional* and *marginal* R^2 suggests that the random effects significantly help to explain the dependent variable.

We use the values of the estimated coefficients of independent variables generated by the three models to measure the extent to which independent and dependent variables are correlated. The estimated coefficients can be positive or negative. A negative coefficient indicates that the variable has a direct relationship with project popularity. A positive coefficient indicates that the variable has an inverse relationship with project popularity. We use the $+ve$ or $-ve$ sign of the estimated coefficient of each variable to produce *upward* and *downward* arrows, respectively. The *upward* and *downward* arrows represent a direct or inverse relationship, respectively, between an independent variable and project popularity. We use the odds ratios [2] to measure the association of the dependent variable with the presence/absence of a binary independent variable (or the increase/decrease of a continuous independent variable) while holding the other variables at a fixed value. For example, odds ratios can explain how project popularity differs between projects that have larger team sizes and projects that have smaller team sizes. We compute the odds ratios by taking the exponentiation of the estimated coefficients obtained from the model for each independent variable. For categorical independent variables, the odds ratio of each categorical level is computed over the reference level. For example, the odds ratio of the *Difficulty* factor is computed over the reference level of *Difficulty* (i.e., the '*Easy*' level).

Findings. Table 4 shows the performance of the linear mixed-effects models in terms of *marginal* R^2 and *conditional* R^2 values. Our results indicate that the models maintain a better performance with the use of the *Platform* and *Age* of projects as random intercepts. Our models have a low risk of overfitting, since the EPV value computed for our dataset is 50. We observe that the *respects* model obtains the highest goodness-of-fit among the other two models (i.e., an R^2 of 0.63). We also observe that the *respects* model is not very sensitive to the variance of the project *Platform* and *Age* as compared to the *views* and *trending* models (i.e., the conditional R^2 of the *respects* model improves by 10%). Nevertheless, the random intercepts improve the performance of the *views* and *trending* models by 30% and 26%, respectively.

Table 4: Performance of the linear mixed-effects models

Model	Marginal R^2	Conditional R^2
Views model	0.26	**0.56**
Respects model	0.53	**0.63**
Trending model	0.34	**0.60**

Table 5 shows the variable importance results obtained from fitting the three linear mixed-effect models. Variables are descendingly sorted by the χ^2 values of the Views model. For each independent variable, we show its estimated coefficient, its χ^2 value, the *p-value* (represented by $Pr(<\chi^2)$), its significance to model the popularity of IoT projects, and whether each independent variable has a direct *upward* or inverse *downward* association with the project popularity.

Project Description

The number of channels and tags of an IoT project have a significant association with the popularity of the project. Project owners can associate their projects to several channels in the Hackster community. Channels can either be platforms that are used by the project or other community channels (e.g., *Women in Hardware*). Users who join the Hackster community can identify the channels that they belong to. Hence, whenever a project related to one of the channels is published, all users can be notified about the project in their feeds. In addition, users can open their channels at any time to explore the newly published projects related to those channels. Similar to the related channels, project owners can also assign a number of tags to their projects. We observe that the number of tags also shares significant importance to model the popularity of IoT projects. Prior research has also reported the vital importance of tags in attracting users [66]. Therefore, project owners should consider assigning as many relevant tags and channels to the projects, since the number of tags and channels is highly associated with the three popularity measures of IoT projects.

We observe that featured projects are more popular than other non-featured projects. The Hackster team marks certain IoT projects as featured projects. Our dataset contains 667 projects (i.e., 4% of the active projects) marked as featured projects. Users can access the featured IoT projects

Table 5: Results of the linear mixed-effects model – factors of each group are descendingly sorted by the χ^2 values of the Views model (**bold** variables share a common, significant association in all the three models)

	Variable	*Views* model				*Respects* model				*Trending* model			
		χ^2	$Pr(<\chi^2)$	Signf.+	Rel.	χ^2	$Pr(<\chi^2)$	Signf.	Rel.	χ^2	$Pr(<\chi^2)$	Signf.	Rel.
Project Description	**Number of channels**	247.829	$< 2.2e^{-16}$	***	↗	124.300	$< 2.2e^{-16}$	***	↗	109.990	$< 2.2e^{-16}$	***	↗
	Featured project	154.114	$< 2.2e^{-16}$	***	↗	428.349	$< 2.2e^{-16}$	***	↗	116.369	$< 2.2e^{-16}$	***	↗
	Copyright license	83.609	$1.5e^{-11}$	***	-	111.852	$< 2.2e^{-16}$	***	-	69.005	$6.7e^{-09}$	***	-
	Difficulty level	56.157	$1.9e^{-11}$	***	-	35.323	$4.0e^{-07}$	***	-	58.554	$5.8e^{-12}$	***	-
	Number of story sections	33.911	$5.8e^{-09}$	***	↗	0.687	0.4073		↘	43.663	$3.9e^{-11}$	***	↗
	Project type	32.302	$3.6e^{-05}$	***	-	32.913	$2.7e^{-05}$	***	-	30.127	$9.0e^{-05}$	***	-
	Cover (video)	30.480	$3.4e^{-08}$	***	↗	131.819	$< 2.2e^{-16}$	***	↗	54.495	$1.6e^{-13}$	***	↗
	Won a contest	23.178	$1.5e^{-06}$	***	↗	10.277	0.0013	**	↗	27.510	$1.6e^{-07}$	***	↗
	Number of links in story	17.753	$2.5e^{-05}$	***	↗	2.975	0.0845	.	↗	3.604	0.0576	.	↗
	Number of tags	12.391	0.0004	***	↗	28.763	$8.2e^{-08}$	***	↗	19.219	$1.2e^{-05}$	***	↗
	Number of contests	10.742	0.0010	**	↘	3.464	0.0627	.	↘	31.424	$2.1e^{-08}$	***	↘
	Number of videos	6.809	0.0091	**	↗	138.952	$< 2.2e^{-16}$	***	↗	6.946	0.0084	**	↗
	Number of platform projects	4.411	0.0357	*	↗	5.622	0.0177	*	↗	4.850	0.0277	*	↗
	Description length	2.154	0.1422		↘	17.677	$2.6e^{-05}$	***	↘	0.007	0.9316		↗
	Number of schematics	1.783	0.1818		↘	2.325	0.1273		↗	1.136	0.2864		↘
	Number of images	1.003	0.3165		↗	17.663	$2.6e^{-05}$	***	↗	0.056	0.8128		↘
	Last modified	0.418	0.5181		↗	203.616	$< 2.2e^{-16}$	***	↘	188.107	$< 2.2e^{-16}$	***	↘
	Length of story	0.240	0.6239		↗	5.295	0.0214	*	↗	7.150	0.0075	**	↗
	Estimated time	0.105	0.7462		↗	0.061	0.8045		↗	1.177	0.2781		↗
Hardware	**Most common purchase source**	1465.484	$< 2.2e^{-16}$	***	-	966.541	$< 2.2e^{-16}$	***	-	1308.758	$< 2.2e^{-16}$	***	-
	Vendors per hardware component	551.122	$< 2.2e^{-16}$	***	↗	87.638	$< 2.2e^{-16}$	***	↗	296.974	$< 2.2e^{-16}$	***	↗
	Hardware-controlling code	20.903	$2.9e^{-05}$	***	-	39.475	$2.7e^{-09}$	***	-	33.888	$4.4e^{-08}$	***	-
	Hardware components	20.167	$7.1e^{-06}$	***	↗	3.091	0.0787	.	↗	26.894	$2.2e^{-07}$	***	↗
	Estimated cost mentioned	16.946	$3.8e^{-05}$	***	↗	0.051	0.8220		↘	28.475	$9.5e^{-08}$	***	↗
	Hand & fabrication tools	2.857	0.0910	.	↘	1.474	0.2247		↘	0.174	0.6762		↗
	Tools without links to purchase	2.254	0.1333		↘	1.408	0.2354		↗	18.690	$1.5e^{-05}$	***	↘
	Software applications & services	0.069	0.7925		↗	30.668	$3.1e^{-08}$	***	↗	0.009	0.9235		↗
Project owner(s)	**Geographical location**	246.199	$< 2.2e^{-16}$	***	-	350.799	$< 2.2e^{-16}$	***	-	265.149	$< 2.2e^{-16}$	***	-
	Project owner's communities	35.555	$2.5e^{-09}$	***	↘	0.276	0.5996		↘	27.634	$1.5e^{-07}$	***	↘
	Project owner's followers	32.466	$1.2e^{-08}$	***	↗	230.030	$< 2.2e^{-16}$	***	↗	60.915	$6.0e^{-15}$	***	↗
	Project owner's likes	29.280	$6.3e^{-08}$	***	↗	6.176	0.0129	*	↗	12.770	0.0004	***	↗
	Number of project owners	25.947	$3.5e^{-07}$	***	↘	33.361	$7.7e^{-09}$	***	↗	45.031	$1.9e^{-11}$	***	↘
	Personal Web page available	20.099	$7.4e^{-06}$	***	↗	6.480	0.0109	*	↗	10.776	0.0010	**	↗
	Project owner's channels	11.733	0.0006	***	↗	9.884	0.0017	**	↘	6.950	0.0084	**	↗
	Length of project owner's biography	6.120	0.0134	*	↘	6.894	0.0086	**	↘	16.015	$6.3e^{-05}$	***	↘
	Project owner's contests	4.703	0.0301	*	↘	21.350	$3.8e^{-06}$	***	↗	0.526	0.4682		↗
	Project owner's skills	3.470	0.0625	.	↗	0.731	0.3925		↘	0.290	0.5901		↗
	Project owner's awards	0.258	0.6116		↗	25.886	$3.6e^{-07}$	***	↘	1.682	0.1947		↘
	Project owner's followings	0.139	0.7089		↗	11.380	0.0007	***	↘	2.644	0.1039		↘
	Project owner's respects	0.001	0.9704		↗	0.024	0.8769		↗	0.022	0.8820		↗
Comments	**Number of positive comments**	98.318	$< 2.2e^{-16}$	***	↗	2691.746	$< 2.2e^{-16}$	***	↗	526.692	$< 2.2e^{-16}$	***	↗
	Number of project owner's replies	6.141	0.0132	*	↗	0.016	0.9006		↗	250.055	$< 2.2e^{-16}$	***	↗
	Number of negative comments	0.102	0.7499		↗	2.080	0.1492		↗	4.890	0.0270	*	↘

+Significance codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

using a dedicated web page on Hackster. In addition, the featured projects appear on the feed pages of every community user. Therefore, it is important for project owners to understand the criteria to make their projects featured, since we observe that featured projects are likely to be more popular than other projects. It is also important for Hackster to highlight such criteria to project owners to allow them to improve their projects accordingly.

Participating (but not winning) in online contests is likely to have an inverse relationship with the popularity of IoT projects. Our results indicate that the more contests a project participate in, the less likely for the project to become popular. On the other hand, winning at least one contest is most likely to improve the popularity of the projects. This result indicates that project owners should be more careful about whether to participate in a contest or not if the chances of winning are low. For example, the Internet of Toiletries project[36]

[36] `https://www.hackster.io/aros-automatic-reorder-system/internet-of-toiletries-9d7897`

participated in five contests but did not win any of them. That project has around 1,100 views and only 6 respects. Conversely, the WalaBeer Tank project[37] participated in one contest and won that contest. That project has around 23,000 views and only 157 respects.

Hardware

Providing the code required to control the project hardware has a strong association with project popularity. The odds of having higher views, respects, and trending scores for projects that share their code in a remote repository (e.g., GitHub) are 8−11% higher than the odds for projects that do not share the code. Sharing the code as a single file on Hackster can also increase the odds of making a project highly viewed and trending by 1−9%. However, we could not observe any evidence that a single-file code sharing would increase the number of respects of projects. These results encourage project owners to better use remote repositories when sharing the project code with other users.

The more alternatives to purchase hardware, the more popular are the IoT projects. Our results indicate that projects that provide more purchase options to hardware components are more popular than others. In particular, for every vendor added to purchase a hardware component, the odds for a project to become more popular increase by 6−20%. This result encourages project owners to properly choose the hardware components that can be ordered from different suppliers to give more purchasing flexibility to users.

Project owner(s)

As the number of followers of project owners increases, the popularity of the project increases. We observe a direct strong association of project popularity with the number of followers of project owners. In social media, being followed allows your posts and updates to be seen by a large number of users [14]. Kwak et al. [30] reported that the popularity of a tweet is correlated with the popularity of the tweet writer, which both can be estimated by the number of followers. Therefore, project owners should work hard to build connections and gain more followers. This can be achieved by interacting with projects and posts of other users in the community.

Project popularity is most likely to increase if project owners share links to their personal web pages, write concise biographies, and interact with posts of other users. Sharing links to social media accounts or personal web pages is more likely to increase project popularity than writing lengthy biographies. Our results show that a lengthy biography containing details that are unrelated to the IoT technology has an inverse relationship with project popularity. For example, the project owner of the 'Z-Wave Mouse Trap project' [38] has a 30-word biography, in which he writes about his hobbies and family members. Such information is less likely to be associated with project popularity. In addition, our results indicate that a project may be popular even if the project owners have no biography. Moreover, project owners need to be more actively positive towards the posts of other users, since we observe that projects are more likely to be popular if the project owners give more likes to projects or posts of other users.

User feedback

Positive comments have a significant relationship with the popularity of IoT projects. Our results indicate that projects that receive positive feedback from the IoT community are more popular than other projects. An increase in the number of pleasant user comments increases the odds of making a project highly viewed, respected, and trending by 13%, 71%, and 31%, respectively. In addition, negative user feedback is likely to have a strong inverse association with project trending scores, but no association with the project views and respects could be observed.

As project owners constantly reply to the concerns raised by the community users, project popularity is most likely to increase. We observe a strong association between both project views and trending scores and the number of replies provided by project owners to user comments. However, we could not observe strong evidence that the replies of project owners have an association with the respects received by the project. We hypothesize that such a weak association is a result of the fact that users might need much more than a reply to change their opinions about a project (i.e., actual changes to the project implementation would be more appreciated by users). In addition, our models reveal that positive comments have a higher association with project respects. Typically, users who like a project are less likely to leave negative comments, and replying to such comments is less likely to increase the project respects. Project owners are more likely to

[37] `https://www.hackster.io/aros-automatic-reorder-system/internet-of-toiletries-9d7897`

[38] `https://www.hackster.io/eradicatore/z-wave-mouse-trap-d3bcb6`

reply to those comments that are negative or asking for clarification, which we observe that such comments have a weak association with the popularity of projects as well. As a result, our regression model finds no significant evidence that replies to comments would eventually lead to project respects.

To understand the relationship between replies to comments and the respects of a project, we perform a more in-depth analysis on two sample projects. First, we investigate the 'Arduino Web Editor',[39] a project with the highest number of respects in our dataset. We observe that this project has received over 100 comments, but the project owner has responded to only one comment. Second, we investigate the 'MATRIX Voice and MATRIX Creator Running Alexa'[40] project. Such a project has a lower number of respects (i.e., less than 20). Yet, the owners of such a project have replied to 50% of the user comments. As a result of such cases, our logistic regression model finds no strong association between the number of replies to user comments and the number of respects received by a project.

> *All groups of factors contribute to the popularity of IoT projects. The most important factors, such as project channels and tags, hardware purchasing practices, profile and followers of project owners, and positive user feedback, share common associations with the project views, respects, and trending scores. However, important factors may have inconsistent associations with project popularity.*

4.3 ***RQ$_3$*: *What are the inconsistent associations between the factors and popularity measures?***

Motivation. The results of RQ2 suggest that important factors may have disagreeing associations with the popularity of IoT projects. In this RQ, we aim to understand the factors that may have a significant association with one or two popularity measures (say *views* and *trending scores*) but have an insignificant association with the other popularity measure(s) (say *respects*). In addition, we discuss the significant factors that may have a direct relationship with one or two popularity measures but have an inverse relationship with the other popularity measure(s).

Approach. We perform subsequent analyses on the estimated coefficients of the independent variables that have conflicting relationships with project popularity. We use the χ^2 values and the computed odds ratios of the variables in the different models to understand the varying associations with project popularity.

Findings. *Intermediate and Advanced projects are more likely to gain better popularity than Showcase projects.* Analyzing the coefficients of different difficulty levels shows that sophisticated projects are more popular (in terms of views, respects, and trending scores) than other projects. As an exception, the *Expert* difficulty level is strongly associated with the project respects only (i.e., there is no strong evidence that the *Expert* level is associated with highly viewed and trending projects). On the other hand, we observe that *Showcase* projects are less popular than any other projects. Conversely, *Getting started* projects are more likely to be highly respected than any other projects. Therefore, project owners should take into consideration that specifying the type and difficulty level of a project is crucial and should be handled carefully and not in an ad hoc manner.

The project copyright license has a significant relationship with the popularity of IoT projects. Our results indicate that licensing an IoT project is strongly associated with project popularity. We show in Table 6 the estimated coefficients obtained from the linear mixed-effects models for each *copyright* license. We observe that only the `MIT`, `CC BY-NC-SA`, and `CC BY-SA` licenses share significant importance to model the popularity of IoT projects among the three popularity measures. The other copyright licenses have different associations among the three popularity measures. For example, the '`GPL3+`', '`CC BY-NC`', and '`CC BY`' licenses are significantly associated with the *views* and *trending scores* popularity measures, whereas no association could be observed with the project *respects*. The *respects*-based popularity has a distinguished strong association with the `SHL` and `CC BY-ND` licenses. Similar to our observation, Sen [48] observed a strong relationship between license types and the popularity of FLOSS projects. Therefore, project owners should keep the restrictions of the available licensing mechanisms in mind prior to licensing their projects.

Projects with larger team sizes have more respects but less views and trending scores. However, larger numbers of channels connected to project owners have a contrary association. Our results indicate that the odds of giving a *thumbs-up* to an IoT project increases by the increase of the number of development team members. However, larger team sizes have an inverse association with the number of views and tending scores of the projects, since users can give the *thumbs-up* without even opening the project

[39] `https://www.hackster.io/Arduino_Genuino/getting-started-with-arduino-web-editor-on-various-platforms-4b3e4a`

[40] `https://www.hackster.io/matrix-labs/matrix-voice-and-matrix-creator-running-alexa-c-version-9b9d8d`

Table 6: Estimated coefficients obtained from the linear mixed-effects model for each *Copyright* license

License	*Views* model				*Respects* model				*Trending scores* model			
	Coef.	$Pr(<\chi^2)$	Signf.	Rel.	**Coef.**	$Pr(<\chi^2)$	Signf.	Rel.	**Coef.**	$Pr(<\chi^2)$	Signf.	Rel.
MIT	−0.094	< 0.001	***	↗	0.068	< 0.001	***	↘	−0.052	0.015	*	↗
Apache-2.0	−0.116	< 0.001	***	↗	−0.004	0.881		↗	−0.020	0.467		↗
GPL3+	−0.135	< 0.001	***	↗	−0.028	0.060	.	↗	−0.103	< 0.001	***	↗
LGPL	−0.155	< 0.001	***	↗	0.014	0.584		↘	−0.078	0.007	**	↗
CC BY-NC	−0.158	0.001	***	↗	−0.081	0.032	*	↗	−0.083	0.058	.	↗
CC BY	−0.173	< 0.001	***	↗	−0.037	0.162		↗	−0.102	0.001	***	↗
CC BY-NC-SA	−0.180	< 0.001	***	↗	−0.128	< 0.001	**	↗	−0.128	< 0.001	***	↗
CC BY-SA	−0.139	0.001	**	↗	−0.076	0.031	*	↗	−0.156	< 0.001	***	↗
CERN-OHL	−0.274	0.002	**	↗	0.030	0.672		↘	0.008	0.923		↘
TAPR-OHL	−0.269	0.022	*	↗	0.112	0.241		↘	0.016	0.885		↘
CC BY-NC-ND	−0.115	0.089	.	↗	0.141	0.010	*	↘	−0.168	0.008	**	↗
CC0	−0.153	0.098	.	↗	−0.135	0.074	.	↗	−0.069	0.428		↗
SHL	−0.361	0.352		↗	0.667	0.037	*	↘	0.652	0.073	.	↘
CC BY-ND	0.019	0.870		↘	0.355	< 0.001	***	↘	0.083	0.431		↘
MPL-2.0	0.303	0.114		↘	0.130	0.406		↘	0.310	0.085	.	↘

page. On the contrary, a project is likely to be viewed considerably and become trending if the project owners are connected to a larger number of community channels. However, the odds of giving a *thumbs-up* to an IoT project decreases as the number of channels increases.

Projects are likely to be highly viewed and trending if more hardware components are used or an approximate cost of the hardware is indicated. It could be surprising to know that more hardware components can increase the popularity of IoT projects. However, it is important to note that projects may not list the exact number of hardware components required, which can have an inverse relationship with project popularity. It is also important to note that having more hardware components does not always indicate higher costs. For example, a project with 10 hardware components (e.g., capacitors and resistors) may cost less than a project that requires only 2 hardware components (e.g., a digital camera and sensor). We were unable to obtain information about hardware prices due to the variety of hardware suppliers, quantities, and currencies. Nevertheless, we observe that projects that mention an approximate cost of all (or part) of hardware components are 10−12% more popular than other projects. We do note that we could not observe any evidence that such two factors have a relationship with the number of respects of IoT projects.

We observe a significant association of* 19 *suppliers of hardware components with the three IoT project popularity measures. Projects in our dataset contain over 470 different hardware suppliers. 176 of these suppliers have significant associations with at least one popularity measure. In Table 7, we show the suppliers that share a common significant association among the three project popularity measures. We observe that, the more frequent the `farnell`, `seeedstudio`, `erlerobotics`, or `hologram` websites are used to purchase hardware components, the more popular are the projects. However, projects that use `xinabox` more frequently are less popular than other projects. Despite such common associations, we observe that some suppliers have conflicting associations with project popularity. For example, highly viewed and trending projects are associated with using `adafruit`, `amazon`, `sparkfun`, and `microsoft` to purchase hardware components. However, those projects are likely to be *less* respectful if such suppliers are selected to purchase hardware components from. We hypothesize that the small number of respects that is associated with the use of such suppliers is most likely due to higher prices or prolonged delivery processes of such projects.

Licensing a project has a direct relationship with the number of views and trending scores of IoT projects but has an inverse relationship with the respects of IoT projects (e.g., the MIT license). Projects that use general websites to purchase hardware (e.g., Amazon or Microsoft) are likely to be highly viewed and trending (but not respected).

5 Discussion

In this section, we discuss our findings on the important factors in terms of direct implications for IoT project owners and Hackster.

5.1 Project owners

Project owners should be aware of the fact that having more views is not always enough to indicate that a project is popular. Highly viewed projects should also be well-perceived by the community users.

Table 7: Estimated coefficients obtained from the linear mixed-effects model for each *Common Hardware Supplier*

Hardware supplier	*Views* model				*Respects* model				*Trending scores* model			
	Coef.	$Pr(<\chi^2)$	Signf.	Rel.	**Coef.**	$Pr(<\chi^2)$	Signf.	Rel.	**Coef.**	$Pr(<\chi^2)$	Signf.	Rel.
adafruit	−0.247	< 0.001	***	↗	0.078	0.025	*	↘	−0.239	< 0.001	***	↗
amazon	−0.314	< 0.001	***	↗	0.161	0.012	*	↘	−0.451	< 0.001	***	↗
sparkfun	−0.319	< 0.001	***	↗	0.130	0.005	**	↘	−0.165	0.002	**	↗
farnell	−0.375	< 0.001	***	↗	−0.688	0.001	***	↗	−0.697	0.006	**	↗
seeedstudio	−0.394	< 0.001	***	↗	−0.438	0.040	*	↗	−0.867	< 0.001	***	↗
modmypi	−0.503	< 0.001	***	↗	0.413	0.008	**	↘	−0.537	0.003	**	↗
patternagents	−0.637	< 0.001	***	↗	0.143	0.001	**	↘	−0.191	< 0.001	***	↗
hologram	−0.652	< 0.001	***	↗	2.852	¡0.001	***	↘	3.662	$< 2e-16$	***	↘
matrix	−0.750	< 0.001	***	↗	−0.107	0.037	*	↗	−0.563	$< 2e-16$	***	↗
aiyprojects	−1.002	< 0.001	***	↗	0.103	0.005	**	↘	−0.163	< 0.001	***	↗
erlerobotics	−1.002	< 0.001	***	↗	−0.931	¡0.001	***	↗	−0.831	< 0.001	***	↗
reference	−2.050	< 0.001	***	↗	0.405	0.003	**	↘	−0.312	0.050	*	↗
harborfreight	2.007	< 0.001	***	↘	0.102	0.012	*	↘	−0.289	< 0.001	***	↗
lattepanda	−0.394	0.009	**	↗	−0.170	0.018	*	↗	−0.497	< 0.001	***	↗
bauhaus	−1.122	0.005	**	↗	0.081	0.023	*	↘	−0.232	< 0.001	***	↗
xinabox	0.535	0.001	**	↘	0.356	0.002	**	↘	0.332	0.027	*	↘
dexterindustries	−0.262	0.032	*	↗	0.619	0.043	*	↘	−0.813	0.020	*	↗
microsoft	−0.437	0.022	*	↗	0.868	¡0.001	***	↘	−0.568	< 0.001	***	↗
parrot	−0.539	0.038	*	↗	−0.195	0.002	**	↗	−0.474	< 0.001	***	↗

Profile of the project owner. Project owners are encouraged to write short (but elegant) biographies, since we observe that a shorter biography is associated with highly popular IoT projects. Instead of writing much about themselves, project owners are encouraged to provide links to their personal pages, since we observe that doing so is associated with project popularity. In addition, project owners should be more active with the posts and other channels in the community, since we observe that project owners who are more involved with the community (e.g., give respects to other projects/posts) are associated with popular projects. In addition, being more active will most likely increase the number of followers to project owners, which we also observe that it shares a strong association with project popularity.

Participation in contests. It is better for project owners to not participate in contests if the chances of winning are low, since we observe an inverse relationship between participating in (but not winning) a contest with project popularity.

5.2 Hackster

Hackster provides project owners with initial guidelines on how to properly write a page for an IoT project. Our results can be used to improve the guidelines provided by Hackster.

Featured & trending projects. Hackster guidelines should make the criteria or practices that make a project featured clear to project owners. It is also important to describe the *trending* in the guidelines to allow project owners to compete for ranking their project up in the trending-based project listings.

Project cover. The guidelines provided by Hackster are mostly related to improving the page of a project. The guidelines indicate that the cover picture of a project should be of high resolution and should show the end result of the project. Based on our results, the guidelines should also indicate that using a short video instead of a cover picture can help to attract more people to engage with the project.

Tags & channels. The guidelines of Hackster indicate that tags should be limited to a maximum of 3 and that project owners should avoid using tag descriptions (e.g., Arduino or Raspberry Pi). Nonetheless, our results reveal that as the number of tags increases, the popularity of a project is likely to increase. In addition, we observe that connecting projects to many more related channels is strongly associated with the project popularity.

Code. The current guidelines of Hackster are very brief when it comes to the code of a project. The guidelines indicate that the proper language should be selected for code files and that no placeholders should be used. According to our results, publishing the hardware-controlling code in a remote repository is strongly associated with project popularity.

The guidelines should encourage project owners to avoid uploading their code as single or zip files and use code repositories instead.

Profiles & activities of project owners. The current guidelines of Hackster are oblivious to the actions or profiles of project owners. The guidelines could be improved by including our empirical observations regarding the aspects involving project owners. For example, mentioning that project owners should promptly reply users could improve the guidelines.

6 Threats to Validity

In this section, we discuss the potential threats to the validity of our work.

6.1 Construct Validity

Construct threats to validity are concerned with the degree to which our analyses measure what we claim to analyze [50]. In our study, we rely on the data collected from Hackster. Mistakenly computed values can have an influence on our results. However, we carefully filter and test the data to reduce the possibility of wrong computations that may impact the analyses in this paper. In addition, some of the factors used as independent variables in our models may not be actionable for project owners (e.g., the project owners have almost no control over the number of followers). However, we keep such factors as control variables in our models [40]. Future work could deeply investigate our observations and improve the body of knowledge about the popularity of IoT projects.

6.2 Internal Validity

Internal threats to validity are concerned with the ability to draw conclusions on the relation between the independent and dependent variables [50]. We study 57 factors. However, we are aware that these factors are not fully comprehensive and using other factors may affect our results. For example, the response time of project owners can be another explanatory metric. However, Hackster presents the response dates with varying granularity levels, including day, weak, month, and year; as the dates get older, the granularity increases. For example, 24% of our collected comments are dated as *"a year ago"*. Therefore, we could not include the response time in our study. Future work may study more factors to explain project popularity better. In our correlation analysis, deciding which variables to keep in the linear mixed-effects models may have an impact on the results of the models. To make our obtained results reproducible, we explicitly define our selections of variables for all possible pairs of highly correlated variables. Moreover, activities by Hackster may influence the popularity of IoT projects. For example, a project can be labeled as *beginners*, *intermediate*, *advanced*, or *experts* by Hackster, and can be featured in a special page on Hackster. We include such metrics in our models and find that they are associated with the popularity of IoT projects. Nevertheless, we have no control over any other unforeseen activities that might be performed by Hackster, since we rely on the metrics that we collect from the web pages of projects and project owners.

6.3 External Validity

External threats are concerned with our ability to generalize our results [50]. Our study is based on 15,007 active projects collected from the Hackster online community. Therefore, we cannot generalize our conclusions to IoT projects in other online communities (e.g., Instructables[41]). Future work should investigate whether our observations may hold for projects published in other online communities.

7 Related Work

In this section, we present the existing work related to IoT technology and the popularity gained from online communities.

7.1 IoT studies

Researchers have studied IoT in a wide variety of problems, i.e., context-aware IoT approaches [4,5,11,16,27,33,38,43], fault-tolerance in IoT services [46,53,69], IoT and cloud computing [10,21,28,45,67], and IoT service composition [7,15,18,20,23,25,41,53,59,68].

Chattopadhyay et al. [11] presented an analytical method that helps engineers to build IoT applications without the need to have heavy knowledge of signal processing or any other specific domains. D'Oca and Hong [16] proposed a framework with two data mining techniques (i.e., clustering and associated rules) to identify the behavior of occupants related to the opening and closing of windows. The authors found that indoor air temperature, outdoor air temperature, and the presence of occupants were the most important factors for the opening of windows. As for window closing, the indoor air temperature, and outdoor air temperature are the most important factors. Regarding fault tolerance

[41] `https://www.instructables.com/`

in IoT services, Su et al. [53] proposed the Strip approach, which allows the achievement of *failover* mechanisms upon the replacement of IoT devices. The results of their research show that failures may be recovered within seconds without the need for project owners and administrators in the process.

Botta et al. [10] conducted a literature review to understand the potential applications and challenges of using IoT and cloud computing together (i.e., the *CloudIoT* paradigm). The authors identified several open issues, such as the need for standardization in both IoT and cloud computing fields. Finally, with respect to IoT composition, Tzortzis and Spyrou [59] proposed a semi-automatic approach that allows project owners to discover, consume, and interconnect IoT services to create more complex services. They evaluated their approach by interconnecting simple IoT-enabled services.

Ustek-Spilda et al. [60] studied how active are social media (in particular, Twitter) discussions about the IoT technology in in Europe. The authors found that users from the same geographical context are more likely to be connected online than users from different geographical contexts. The authors also observed that IoT-related hashtags (e.g., #healthcare, #hardware, #IoT, and #startups) are highly correlated.

Unlike the aforementioned work, our study focuses on investigating the factors that share a significant relationship with the popularity of IoT projects rather than approaches that can improve the IoT technology.

7.2 Popularity studies

Studies in the literature have investigated the factors that share a significant relationship with the popularity of software projects on GitHub [1,3,8,9,13,49,62,70] and mobile applications and data services [39,54,57, 64].

Consentino et al. [13] summarized the factors that impact the popularity of GitHub projects. Similarly to our observations, Consentino et al.noted that proper documentation and involvement of popular users contribute to the popularity of IoT projects. The importance of documentation in the popularity of GitHub projects has also been stressed by Aggarwal et al. [1]. Weber et al. [62] studied a large set of features that characterize open source projects, including both in-code features and metadata features. Borges et al. [8] used multiple linear regression to study the main factors that have an association with the number of stars of GitHub projects. These factors include the programming languages, application domains, and new features of these projects. Sheoran et al. [49] studied the popularity of GitHub projects in terms of the number of *watchers*. Zhu et al. [70] considered the number of *forks* as a measure of the popularity of GitHub projects instead of the number of *watchers*. Alsmadi and Alazzam [3] used the number of *downloads* as a popularity indicator of GitHub projects. Borges et al. [9] conducted a survey on Stack Overflow users to elicit their opinion about popularity indicators of GitHub projects. The survey results show that *stars* are the most useful measure for the popularity of software projects hosted on GitHub.

Tian et al. [57] investigated the most important factors regarding the ratings of free Android applications. Syer et al. [54] revisited prior empirical findings in software engineering for 15 popular mobile apps. The authors found that the number of core developers in mobile app projects is usually smaller than large desktop/server applications such as the Apache HTTP server. Noei et al. [39] studied the trends of Android mobile apps ranking. The authors found that taking into consideration the user reviews to improve mobile apps helps to improve the ranking of a mobile app. Ye et al. [64] studied the popularity of mobile data services and found that online reviews have a strong influence on the popularity of such services. Tam et al. [55] studied the factors of continuance intention of users to use mobile apps and found that satisfaction and performance expectancy highly influence the intentions of individuals.

In comparison with the aforementioned work, our study is the first to study the popularity of IoT projects using four groups of factors, which are the *project description*, *hardware*, *project owner(s)*, and *user feedback* factors. Our study is important because IoT projects have different characteristics from software projects—IoT projects operate mostly on embedded systems, which implies a lower level of coding when compared to software projects.

Previous studies on the popularity of software and mobile applications rely heavily on code and process factors, user reviews, and other social factors. While some of the factors (e.g., user feedback) used in our study may have also been used in prior research, we believe that it is important to investigate whether such factors share similar or contradicting relationships with the popularity of IoT projects. In addition, our study complements previously investigated factors by involving new factors that particularly capture the characteristics of IoT projects. In particular, we study 12 hardware-related factors, including the number of hardware components, hand tools, project replication time, and hardware suppliers. Such factors have demonstrated a strong association with the popularity of IoT projects. For example, the locations of project owners can influence the availability or delivery delay of hardware components to users located in other parts of the world. Such a factor may be associated with the popularity of IoT projects, since users may be interested in projects in which hardware can be purchased from local suppliers. Moreover, owners of IoT projects may participate in online contests and may win prizes. Considering that contest-winning projects are advertised to other community users, such a factor appeared to have a direct relationship with the popularity of IoT projects.

8 Conclusion

Online IoT communities have emerged as a great communicative platform for IoT practitioners to discuss technical issues and promote IoT projects to potential users for commercial values. Consequently, it is of great interest for IoT project owners to understand the most important factors that share a significant relationship with the popularity of IoT projects. We explore four groups of factors comprising 57 explanatory factors. We conduct an exploratory study on $15,007$ projects that are hosted on Hackster. We observe that all the groups of factors (i.e., *Description*, *Hardware*, *Project owners*, and *Feedback*) make a significant contribution to explain the popularity of IoT projects. Nevertheless, different popularity measures may have different associations with the factors that describe IoT projects. In particular, we observe the following:

– Project views evolve differently from project respects.
– There are platforms, such as Arduino and Raspberry Pi, that are widely used by a large number of projects.
– Assigning more tags and related channels to a project is most likely to increase project popularity.
– Participation in (but not winning) contests is likely to have an inverse relationship with the popularity of IoT projects.
– Projects with a clear estimate of replication costs are highly viewed and trending in the IoT community.
– Using general websites (e.g., Amazon or Microsoft) to purchase hardware is associated with the high number of views and trending scores (but not respects) of IoT projects.
– Projects are more popular if their owners are active in the IoT community, have concise biographies, and acquire more followers.
– Projects with smaller team sizes are highly viewed and trending, but less respectful.
– Positive use feedback on a project has a significant direct association with project popularity.
– Projects are highly viewed and trending if project owners actively reply to user comments.

Project owners can benefit from our observations to improve the popularity of their IoT projects to foster more business opportunities. In addition, Hackster.io can leverage our findings to improve the information provided in the guidelines to allow (new) project owners to improve the popularity of IoT projects. In the future, we plan to expand our study to investigate more online communities (e.g., *instructables.com*) to check whether our observations may hold. We also aim to perform a qualitative study to investigate the current practices that project owners use to improve the popularity of their projects. Moreover, we aim to model featured projects and contest-winning projects to understand the factors that distinguish these projects from other projects.

References

1. Aggarwal, K., Hindle, A., Stroulia, E.: Co-evolution of project documentation and popularity within github. In: Proceedings of the 11th Working Conference on Mining Software Repositories, pp. 360–363. ACM (2014)
2. Agresti, A.: Tutorial on modeling ordered categorical response data. Psychological bulletin **105**(2), 290 (1989)
3. Alsmadi, I., Alazzam, I.: Software attributes that impact popularity. In: 2017 8th International Conference on Information Technology (ICIT), pp. 205–208. IEEE (2017)
4. Barnaghi, P., Sheth, A.: On searching the internet of things: Requirements and challenges. IEEE Intelligent Systems **31**(6), 71–75 (2016)
5. Bauman, K., Tuzhilin, A.: Discovering contextual information from user reviews for recommendation purposes. In: CBRecSys@ RecSys, pp. 2–9 (2014)
6. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. KDD workshop **10**(16), 359–370 (1994)
7. Billet, B., Issarny, V.: From task graphs to concrete actions: a new task mapping algorithm for the future internet of things. In: 11th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), pp. 470–478. IEEE (2014)
8. Borges, H., Hora, A., Valente, M.T.: Understanding the factors that impact the popularity of github repositories. arXiv preprint arXiv:1606.04984 (2016)
9. Borges, H., Valente, M.T.: What's in a github star? understanding repository starring practices in a social coding platform. Journal of Systems and Software **146**, 112–129 (2018)
10. Botta, A., De Donato, W., Persico, V., Pescapé, A.: On the integration of cloud computing and internet of things. In: International Conference on Future Internet of Things and Cloud (FiCloud), pp. 23–30. IEEE (2014)
11. Chattopadhyay, T., Banerjee, S., Maiti, S., Dey, S., Jaiswal, D., Barik, B.: Way to make ourselves redundant: A semantic framework for automated workflow generation for iot. TCS Technical Architects (2015)
12. Chen, M., Liu, X.: Predicting popularity of online distributed applications: itunes app store case analysis. In: iConference, pp. 661–663. ACM (2011)
13. Cosentino, V., Izquierdo, J.L.C., Cabot, J.: A systematic mapping study of software development with github. IEEE Access **5**, 7173–7192 (2017)

14. Crawford, K.: Following you: Disciplines of listening in social media. Continuum **23**(4), 525–535 (2009)
15. De, S., Elsaleh, T., Barnaghi, P., Meissner, S.: An internet of things platform for real-world and digital objects. Scalable Computing: Practice and Experience **13**(1), 45–58 (2012)
16. D'Oca, S., Hong, T.: A data-mining approach to discover patterns of window opening and closing behavior in offices. Building and Environment **82**, 726–739 (2014)
17. Domingos, P.: A few useful things to know about machine learning. Communications of the ACM **55**(10), 78–87 (2012)
18. Eisenhauer, M., Rosengren, P., Antolin, P.: Hydra: A development platform for integrating wireless devices and sensors into ambient intelligence systems. In: The Internet of Things, pp. 367–373. Springer (2010)
19. Faraway, J.J.: Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models, vol. 124. CRC press (2016)
20. Florescu, D., Grünhagen, A., Kossmann, D.: Xl: An xml programming language for web service specification and composition. Computer Networks **42**(5), 641–660 (2003)
21. Fox, G.C., Kamburugamuve, S., Hartman, R.D.: Architecture and measured characteristics of a cloud based internet of things. In: International Conference on Collaboration Technologies and Systems (CTS), pp. 6–12. IEEE (2012)
22. Ghaleb, T.A., Da Costa, D.A., Zou, Y.: An empirical study of the long duration of continuous integration builds. Empirical Software Engineering **24**(4), 2102–2139 (2019)
23. Hachem, S., Teixeira, T., Issarny, V.: Ontologies for the internet of things. In: 8th Middleware Doctoral Symposium, p. 3. ACM (2011)
24. Harrell, F.E.: Regression modeling strategies, with applications to linear models, survival analysis and logistic regression. GET ADDRESS: Springer (2001)
25. Huang, Z., Tsai, B.L., Chou, J.J., Chen, C.Y., Chen, C.H., Chuang, C.C., Lin, K.J., Shih, C.S.: Context and user behavior aware intelligent home control using wukong middleware. In: International Conference on Consumer Electronics-Taiwan (ICCE-TW), pp. 302–303. IEEE (2015)
26. Islam, M.R., Zibran, M.F.: Leveraging automated sentiment analysis in software engineering. In: Proceedings of the 14th International Conference on Mining Software Repositories, pp. 203–214. IEEE Press (2017)
27. Jin, J., Gubbi, J., Marusic, S., Palaniswami, M.: An information framework for creating a smart city through internet of things. IEEE Internet of Things Journal **1**(2), 112–121 (2014)
28. Kamilaris, A., Pitsillides, A., Trifa, V.: The smart home meets the web of things. International Journal of Ad Hoc and Ubiquitous Computing **7**(3), 145–154 (2011)
29. Krutz, D.E., Munaiah, N., Meneely, A., Malachowsky, S.A.: Examining the relationship between security metrics and user ratings of mobile apps: a case study. In: International Workshop on App Market Analytics, pp. 8–14. ACM (2016)
30. Kwak, H., Lee, C., Park, H., Moon, S.: What is twitter, a social network or a news media? In: 19th international conference on World wide web, pp. 591–600. ACM (2010)
31. Lawrence, I., Lin, K.: A concordance correlation coefficient to evaluate reproducibility. Biometrics pp. 255–268 (1989)
32. Lewis, A.J.: Mixed effects models and extensions in ecology with R. Springer (2009)
33. Li, L., Li, S., Zhao, S.: Qos-aware scheduling of services-oriented internet of things. IEEE Transactions on Industrial Informatics **10**(2), 1497–1505 (2014)
34. Li, S., Da Xu, L., Zhao, S.: The internet of things: a survey. Information Systems Frontiers **17**(2), 243–259 (2015)
35. Liao, T.W.: Clustering of time series data—a survey. Pattern recognition **38**(11), 1857–1874 (2005)
36. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability **1**(14), 281–297 (1967)
37. Nakagawa, S., Schielzeth, H.: A general and simple method for obtaining R2 from generalized linear mixed-effects models. Methods in Ecology and Evolution **4**(2), 133–142 (2013)
38. Nambi, S.A.U., Sarkar, C., Prasad, R.V., Rahim, A.: A unified semantic knowledge base for iot. In: Internet of Things (WF-IoT), 2014 IEEE World Forum on, pp. 575–580. IEEE (2014)
39. Noei, E., Da Costa, D.A., Zou, Y.: Winning the app production rally. In: Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 283–294. ACM (2018)
40. Noru, M.J.: IBM SPSS Statistics 19 Guide to Data Analysis. Prentice Hall (2012)
41. Patel, P., Cassou, D.: Enabling high-level application development for the internet of things. Journal of Systems and Software **103**, 62–84 (2015)
42. Peduzzi, P., Concato, J., Kemper, E., Holford, T.R., Feinstein, A.R.: A simulation study of the number of events per variable in logistic regression analysis. Journal of clinical epidemiology **49**(12), 1373–1379 (1996)
43. Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, D.: Context aware computing for the internet of things: A survey. IEEE Communications Surveys & Tutorials **16**(1), 414–454 (2014)
44. Pinheiro, P.: Linear and nonlinear mixed effects models. r package version 3.1-97. http://cran. r-project. org/web/packages/nlme (2010)
45. Rao, B.P., Saluia, P., Sharma, N., Mittal, A., Sharma, S.V.: Cloud computing for internet of things & sensing based applications. In: 6th International Conference on Sensing Technology (ICST), pp. 374–380. IEEE (2012)
46. Reijers, N., Lin, K.J., Wang, Y.C., Shih, C.S., Hsu, J.Y.j.: Design of an intelligent middleware for flexible sensor configuration in m2m systems. In: Sensornets, pp. 41–46 (2013)
47. Sarle, W.: The varclus procedure. SAS/STAT User's Guide, (1990)
48. Sen, R.: Open source software development projects: determinants of project popularity. Tech. rep., EERI Research Paper Series (2006)
49. Sheoran, J., Blincoe, K., Kalliamvakou, E., Damian, D., Ell, J.: Understanding watchers on github. In: Proceedings of the 11th Working Conference on Mining Software Repositories, pp. 336–339. ACM (2014)
50. Shull, F., Singer, J., Sjøberg, D.I.: Guide to Advanced Empirical Software Engineering. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2007)
51. Singh, K.J., Kapoor, D.S.: Create your own internet of things: A survey of iot platforms. IEEE Consumer Electronics Magazine **6**(2), 57–68 (2017)
52. Stewart, K., Ammeter, T.: An exploratory study of factors influencing the level of vitality and popularity of open source projects. ICIS p. 88 (2002)

53. Su, P.H., Shih, C.S., Hsu, J.Y.J., Lin, K.J., Wang, Y.C.: Decentralized fault tolerance mechanism for intelligent iot/m2m middleware. In: Internet of Things (WF-IoT), 2014 IEEE World Forum on, pp. 45–50. IEEE (2014)
54. Syer, M.D., Nagappan, M., Hassan, A.E., Adams, B.: Revisiting prior empirical findings for mobile apps: An empirical case study on the 15 most popular open-source android apps. In: Conference of the Center for Advanced Studies on Collaborative Research, pp. 283–297. IBM Corp. (2013)
55. Tam, C., Santos, D., Oliveira, T.: Exploring the influential factors of continuance intention to use mobile apps: Extending the expectation confirmation model. Information Systems Frontiers pp. 1–15 (2018)
56. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., Kappas, A.: Sentiment strength detection in short informal text. Journal of the Association for Information Science and Technology **61**(12), 2544–2558 (2010)
57. Tian, Y., Nagappan, M., Lo, D., Hassan, A.E.: What are the characteristics of high-rated apps? a case study on free android applications. In: International Conference on Software Maintenance and Evolution (ICSME), pp. 301–310. IEEE (2015)
58. Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **63**(2), 411–423 (2001)
59. Tzortzis, G., Spyrou, E.: A semi-automatic approach for semantic iot service composition. In: Workshop on Artificial Intelligence and Internet of Things in conjunction with SETN (2016)
60. Ustek-Spilda, F., Vega, D., Magnani, M., Rossi, L., Shklovski, I., Lehuede, S., Powell, A.: A twitter-based study of the european internet of things. Information Systems Frontiers pp. 1–15 (2020)
61. Vandekerckhove, J., Matzke, D., Wagenmakers, E.J.: Model comparison and the principle. In: The Oxford handbook of computational and mathematical psychology, vol. 300. Oxford Library of Psychology (2015)
62. Weber, S., Luo, J.: What makes an open source code popular on git hub? In: International Conference on Data Mining Workshop (ICDMW), pp. 851–855. IEEE (2014)
63. Winter, B.: A very basic tutorial for performing linear mixed effects analyses. arXiv preprint arXiv:1308.5499 (2013)
64. Ye, H.J., Chua, C.E.H., Sun, J.: Enhancing mobile data services performance via online reviews. Information Systems Frontiers **21**(2), 441–452 (2019)
65. Yu, C.H.: Exploratory data analysis. Methods **2**, 131–160 (1977)
66. Zarrella, D.: The social media marketing book. " O'Reilly Media, Inc." (2009)
67. Zaslavsky, A., Perera, C., Georgakopoulos, D.: Sensing as a service and big data. arXiv preprint arXiv:1301.0159 (2013)
68. Zhang, Z., Li, X., Liu, C., Su, S., Han, Y.: A service-based approach to situational correlation and analyses of stream sensor data. In: International Conference on Web Services (ICWS), pp. 572–579. IEEE (2017)
69. Zhou, S., Lin, K.J., Na, J., Chuang, C.C., Shih, C.S.: Supporting service adaptation in fault tolerant internet of things. In: 8th International Conference on Service-Oriented Computing and Applications (SOCA), pp. 65–72. IEEE (2015)
70. Zhu, J., Zhou, M., Mockus, A.: Patterns of folder use and project popularity: A case study of github repositories. In: Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, p. 30. ACM (2014)

Authors' Biographies

Taher Ahmed Ghaleb is a Ph.D. candidate at the School of Computing at Queen's University in Canada. Taher is the holder of an Ontario Trillium Scholarship, a highly prestigious award for doctoral students. He has been working as a Research/Teaching Assistant since he obtained his B.Sc. in Information Technology from Taiz University in 2008. His M.Sc. in Computer Science is from King Fahd University of Petroleum and Minerals in 2016. His research interests include empirical software engineering, mining software repositories, machine learning for software testing, and continuous integration.

Daniel Alencar da Costa is a Lecturer (Assistant Professor) at the University of Otago, New Zealand. Daniel obtained his PhD in Computer Science at the Federal University of Rio Grande do Norte (UFRN) in 2017 followed by a Postdoctoral Fellowship at Queen's University, Canada, from 2017 to late 2018. His research goal is to advance the body of knowledge of Software Engineering methodologies through empirical studies using statistical and machine learning approaches as well as consulting and documenting the experience of Software Engineering practitioners.

Ying (Jenny) Zou is the Canada Research Chair in Software Evolution. She is a professor in the Department of Electrical and Computer Engineering, and cross-appointed to the School of Computing at Queen's University in Canada. She is a visiting scientist of IBM Centers for Advanced Studies, IBM Canada. Her research interests include software engineering, software reengineering, software reverse engineering, software maintenance, and service-oriented architecture. More about Dr. Zou and her work is available online at `https://seal-queensu.github.io`

Appendix

(a) Cluster 1

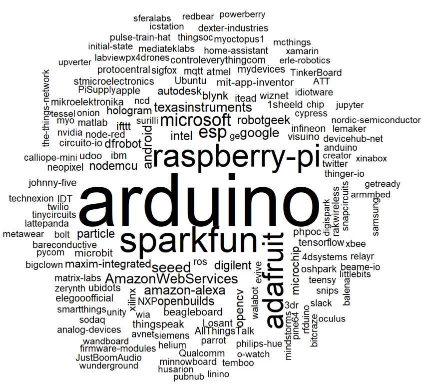
(b) Cluster 2

(c) Cluster 3

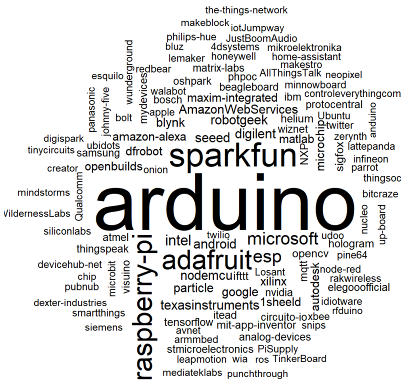
(d) Cluster 4

(e) Cluster 5

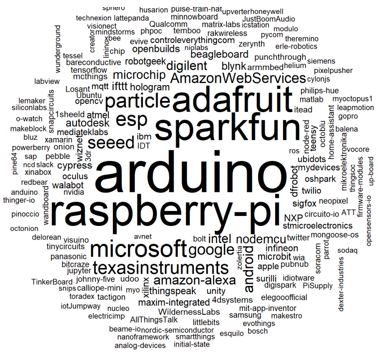
Figure A. Word clouds of the types of projects (tags) in each cluster of daily views



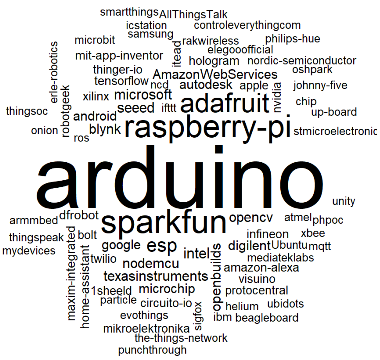
(a) Cluster 1



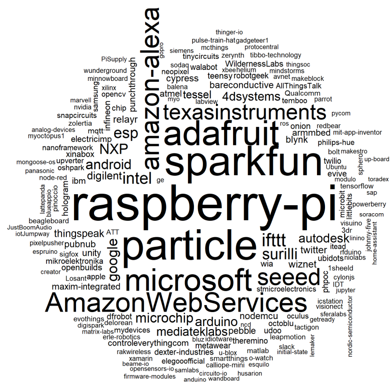
(b) Cluster 2



(c) Cluster 3



(d) Cluster 4



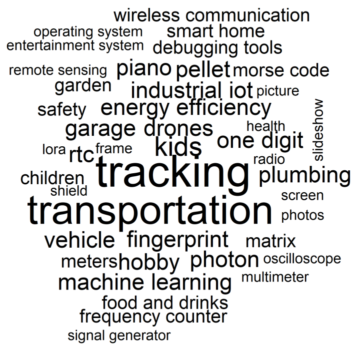
(e) Cluster 5

Figure B. Word clouds of the platforms in each cluster of daily views

(a) Cluster 1

(b) Cluster 2

(c) Cluster 3



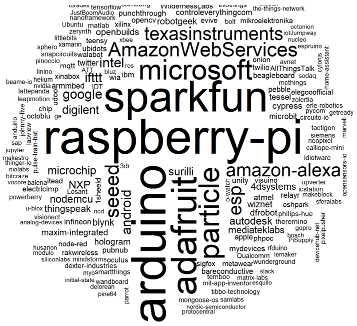
(d) Cluster 4

(e) Cluster 5

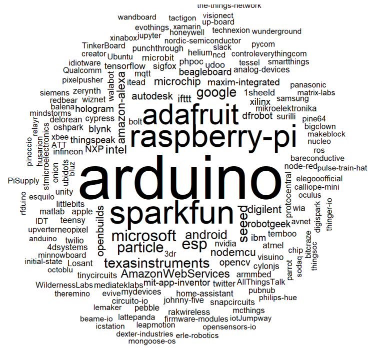
(f) Cluster 6

(g) Cluster 7

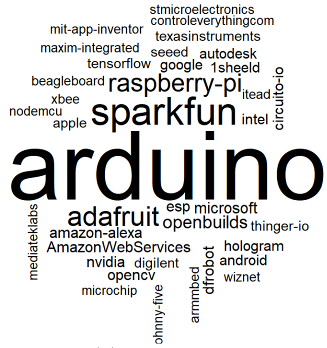
Figure C. Word clouds of the types of projects (tags) in each cluster of daily respects



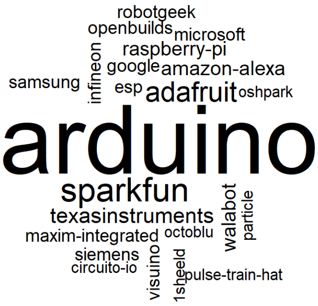
(a) Cluster 1



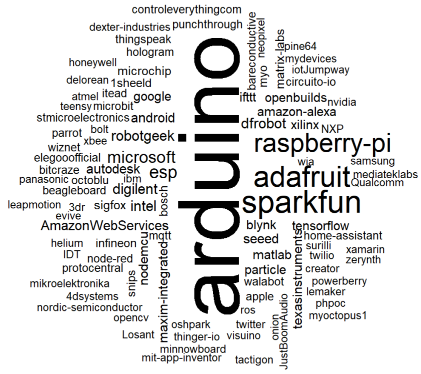
(b) Cluster 2



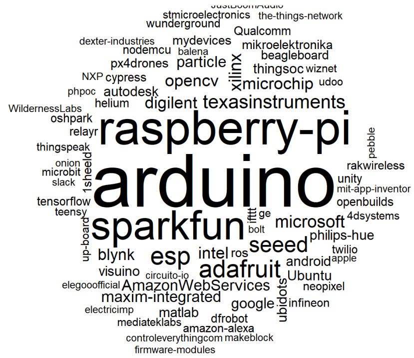
(c) Cluster 3



(d) Cluster 4



(e) Cluster 5



(f) Cluster 6

(g) Cluster 7

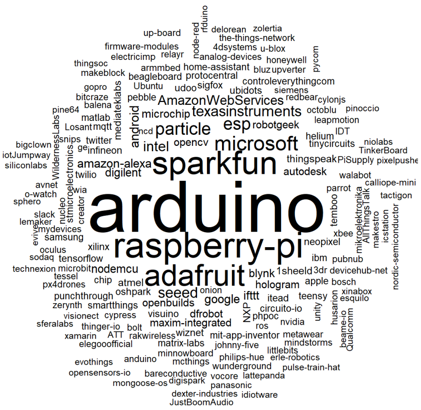
Figure D. Word clouds of the platforms in each cluster of daily respects

(a) Cluster 1

(b) Cluster 2

(c) Cluster 3

Figure E. Word clouds of the types of projects (tags) in each cluster of daily trending scores



(a) Cluster 1

(b) Cluster 2

(c) Cluster 3

Figure F. Word clouds of the platforms in each cluster of daily trending scores