Delft University of Technology

Sustainable Software Engineering
CS4415

Susie: Sustainability Auditor for GitHub Projects

Authors Group 9:
Philippe de Bekker (4876385)
Merel Steenbergen (4784871)
Sebastien van Tiggele (4705165)
Ivor Zagorac (4691202)

April 14, 2023

Contents

1 Introduction

The ICT sector is projected to account for 14% of the global carbon footprint by 2040 [2]. It is the responsibility of software developers to start thinking about sustainability in their projects. The impact on the carbon footprint is mainly environmental sustainability, but there are four more forms of sustainability that are just as important for resilience. Social and individual sustainability focus on people. While social sustainability is more concerned with society and organisations, individual sustainability focuses on the well-being of the people in an organisation and how they interact. Economic sustainability means generating profit such that the project can keep existing. And finally, technical sustainability means that the project is well-maintained.

Existing research on sustainable software engineering has mainly focused on developing guidelines and best practices for building sustainable software from scratch, but there is little information on how to assess the sustainability of already existing projects. By providing an analysis of the sustainability of GitHub repositories, our website can help developers and other stakeholders make more informed decisions about the sustainability of the software they use and contribute to. Additionally, our website can help raise awareness about sustainable software engineering practices and encourage developers to adopt more sustainable habits in their future projects. Even though projects might not be changed anymore, maybe the developing habits still can.

The outline is as follows. Section 2 will provide the details of our proposal. The implementation will be discussed in Section 3, after which the validation and social impact will be explained in Sections 4 and 5. Finally, the conclusion and future work are presented in Section 6 and 7.

2 Proposal

We would like to introduce Susie: A website designed to analyze the social, individual, technical and environmental sustainability, of public GitHub repositories. It allows users to enter the link to a GitHub repository and receive a detailed analysis of its sustainability. This sustainability is measured according to several metrics, which will be explained in Section 2.1. Susie also includes guides concerning sustainable software development, these will be discussed in Section 2.2.

2.1 Metrics

Upon entering the link to a public repository, Susie uses various metrics to generate a sustainability report. This report includes information on the energy consumption of the repository's code, an assessment of the social sustainability of the development practices used in the project and an analysis of the individual sustainability of the developers working on the project. The individual metrics will now be discussed and corroborated.

2.1.1 Programming Languages

Programming languages can vary significantly in terms of energy consumption. As such, understanding the energy consumption of programming languages is crucial to reduce the carbon footprint and promoting sustainable practices in software development. Table 1 includes energy-consumption scores for several programming languages [10]. Susie uses these scores to give advice on the programming language used in a repository. For example, plain JavaScript is about five times more efficient than TypeScript, so repositories with many lines of TypeScript will receive advice to potentially consider JavaScript for future projects.

Programming Language	**Energy Consumption (MJ/LOC)**	**Programming Language**	**Energy Consumption (MJ/LOC)**
C	1	Dart	3.83
Rust	1.03	F#	4.13
C++	1.34	JavaScript	4.45
Ada	1.70	Racket	7.91
Java	1.98	TypeScript	21.50
Pascal	2.14	Hack	24.02
Chapel	2.18	PHP	29.30
Lisp	2.27	Erlang	42.23
Ocaml	2.40	Lua	45.98
Fortran	2.52	Jruby	46.54
Swift	2.79	Ruby	69.91
Haskell	3.10	Python	75.88
C#	3.14	Perl	79.58
Go	3.23		

Table 1: Energy Consumption of Programming Languages. Taken from [10].

2.1.2 Inclusive Language

Inclusive language helps to create a respectful environment for all contributors. This means that certain words and phrases that exclude or marginalize certain people should be omitted, such as gendered pronouns or terms that may be offensive to certain cultures. In a blog post by the Academy Software Foundation, it is explained how to use inclusive language in programming [6]. Susie checks your repository for common terms that may be considered as not inclusive and provides recommendations. Table 2 shows part of the list which is used for inclusive language checks by Susie. Furthermore, Susie checks for profane language using the `bad-words` plugin [14]. Any discovered term is listed with a recommendation and its location in the repository.

Common Terms/Phrases	**Inclusive Suggestions**
master/slave	main/replica, leader/follower, primary/secondary
whitelist	allow list, inclusion list, safe list
blacklist	deny list, exclusion list, block list, banned list
manpower	labor, workforce
guys	folks, people, you all
middleman	middle person, mediator, liaison
he/she, him/her, his/hers	they, them, theirs
crazy/insane	unpredictable, unexpected
normal/abnormal	typical/atypical
grandfather/grandfathering/legacy	flagship, established, rollover, carryover
crushing it/killing it	elevating, exceeding expectations, excelling
owner	lead, manager, expert
sanity check	quick check, confidence check, coherence
dummy value	placeholder value, sample value
native feature	core feature, built-in feature
culture fit	values fit
housekeeping	cleanup, maintenance

Table 2: Some suggestions for inclusive language (not exhaustive), featuring common terms and phrases that may be considered insensitive or exclusionary to certain groups of people.

2.1.3 Governance

Governance is related to certain aspects of an open-source project. This metric examines the presence of a `README.md` file, license, changelog, code of conduct, contributing guidelines, issue template, and pull request template. Based on this, a score between 0 and 100 is calculated. For a visual example, please refer to Figure 11. One additional check that Susie performs is to search for sustainability-related keywords in the `README.md` file. The terms that Susie (currently) takes into account for sustainability are as follows:

- biodegradable
- carbon emission
- carbon footprint
- carbon neutral
- carbon offset
- carbon positive
- circular economy
- climate action
- climate change
- ecological footprint
- ecological impact
- e-waste
- energy consumption
- energy efficiency
- energy saving
- energy statement
- environmental impact
- landfill-free
- leed certification
- organic
- paris agreement
- pollution
- recyclable
- recycling
- sustainability
- sustainable
- waste-to-energy
- waste-to-profit
- zero carbon
- zero waste

If any of these terms are found, the metric indicates that the project addresses sustainability concerns. The aforementioned terms are based on the most popular search terms and glossaries of sustainability terms and definitions [5, 7, 12]. Currently, the terms are handpicked based on intuition, however, for optimal effectiveness, the usage of each individual term should be reviewed by domain experts.

2.1.4 Workflow Analysis

Running workflows locally before executing them remotely is beneficial for saving energy usage. Executing workflows remotely involves utilizing resources such as servers and data centres that require a significant amount of energy compared to running them locally. Therefore, Susie analyses how many workflows have passed and failed in the last hundred runs. If more than 10% of the builds failed, the user will receive a warning that this is quite significant and that the best practice is to always run and fix builds locally before pushing. This number of 10% is quite arbitrary at the moment, but it is a reasonable expectation in larger projects. For a visual overview of the component, please refer to Figure 8.

2.1.5 Issue Sentiment Analysis

Sentiment analysis can find the overall tone of a piece of text. By applying sentiment analysis to issues and merge request comments, we can calculate a sentiment score, which can be very positive, very negative or anything in between. For example, if there are a large number of negative comments on an issue, this may indicate that the issue is causing frustration among the developers and needs to be addressed quickly to avoid problems. Susie uses the npm package `sentiment` to perform sentiment analysis on text data from project issues [11]. It analyzes the text, assigns sentiment scores to individual words, and then uses these scores to determine the overall sentiment of the text. Susie uses this information to provide insights into the sentiment of issues in GitHub repositories.

2.1.6 Contributors

Analysing the contributors of a Github repository can give insights into the distribution of all the work among developers. Susie displays these numbers in a table and a pie chart as a general overview of the contributor component. The bus factor is a statistic of pivotal importance for any project. It is the number of team members whose absence would jeopardize the project (hypothetically speaking: the number of team members hit by a bus). The minimum is 1, and the bigger the number, the better. The metric can be calculated in various ways:

- Pony factor: number of top contributors covering 50% or more of all-time contributions [8].
- Top contributor power: number of contributors as the factor of the biggest contributor.
- Ownership detection: looks at developer aliases and trace change history.

As of now, Susie only implements the first two, as the latter is computationally heavy, a tradeoff that did not add up compared to its added precision.

2.2 Guides

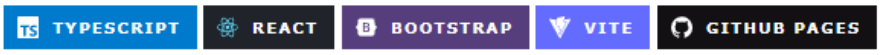
Susie's Guides page (see Figure 2) provides developers with advice on how to make their software more sustainable. The guides cover a range of topics related to sustainability in software development. To introduce developers to sustainability in software, a simple guide about the different types of sustainability with a narrow focus on software development has been created. Another guide, the energy efficiency guide specifically, focuses on the use of programming languages to optimize code for energy efficiency and includes Table 1 - an overview of the energy consumption scores for different programming languages. The goal of this guide is to allow developers to make informed decisions when selecting programming languages for their projects. Lastly, Susie features a guide for inclusive language which provides guidance on how to use language in a way that is respectful, and inclusive, and avoids perpetuating harmful stereotypes or biases as described in Section 2.1.2. Currently, the guides only cover these topics. However, because the Susie project is open source, other developers are encouraged to contribute to the project and add more guides to the page.

2.3 Deliberately Omitted Features

Initially in the design phase, analyzing the test coverage of a project to determine technical sustainability was a desired feature. However, it was quickly discovered that this required cloning every GitHub repository provided to Susie, which is not sustainable in terms of energy consumption and contradicts the goal of the tool. Therefore, it was decided that this feature would be omitted. Some repositories do include a test coverage report, which is something that could be analyzed instead. Essentially, anything can be analyzed as long as the repository does not need to be cloned.

3 Implementation

Susie has been implemented using a modern tech stack that includes TypeScript, React, Bootstrap, and Vite, as well as several additional npm packages that provide functionality such as data validation and visualization. The use of modern web technologies like React and Bootstrap ensures that the site is responsive and user-friendly on a wide range of devices and screen sizes. Furthermore, the React components have been developed in such a way that adding more dashboard sections or new guides can be done effortlessly. Susie has been deployed on GitHub Pages at `https://philippedeb.github.io/susie/`.



The website is organized into several main sections, each of which serves a distinct purpose. The landing page of Susie allows users to browse different sections of the website and prompts users to enter the link to a GitHub repository that they want to analyse, see Figure 1.

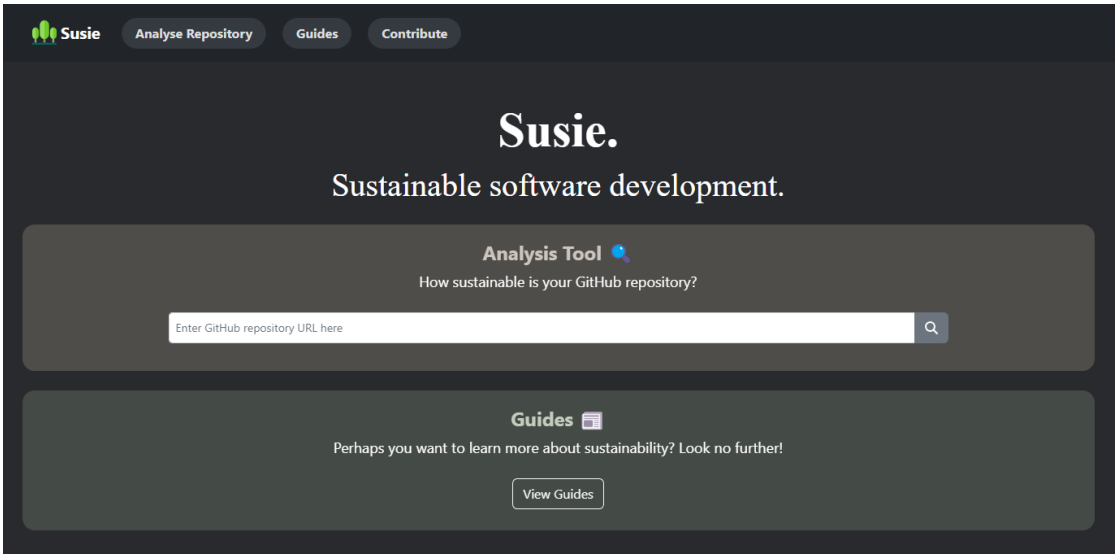


Figure 1: Landing page of Susie. Users can browse different sections and are prompted to enter a link to a GitHub repository to analyse.

Once the analysis is complete, the results are displayed in the dashboard section, which provides information about the sustainability of the repository according to several metrics, as described in Section 2.1. Furthermore, the *Guides* section provides resources and best practices for developers who want to create more sustainable software in the future, as displayed in Figure 2 in Appendix A. Lastly, the *Contribute* button directly links to the open-source GitHub of Susie, encouraging users to help make the impact Susie strives for, you can read more about this in Section 5.

Overall, the Susie website has a clean and intuitive layout that makes it easy for users to navigate and understand. Moreover, to save energy, Susie uses a dark theme throughout the website. For a complete visual overview of Susie, please refer to Appendix A or check out the actual website online. Starring the project on GitHub would also be highly appreciated.

4 Validation

We performed a validation study on a sample of GitHub repositories to assess Susie's ability to identify repositories that incorporate sustainable software engineering practices. Visual data regarding the validation can be found in Appendix B. There are a few fascinating cases to depict regarding the validation of Susie that would otherwise perhaps go unnoticed.

Repositories that Susie performs well on are *Zero Bullshit Haskell* and *GPT4ALL* [1, 9]. In the first one, Susie was successfully able to detect profane language and other non-inclusive language usages, as shown in Figure 18. The latter was correctly deemed as not sustainable. However, Susie should become more aware of big machine learning and artificial intelligence repositories to properly score the sustainability on the impact of such repositories, and not just on keywords in the `README.md` file. Our sentiment analysis in this repository revealed a high rate of false positives due to software engineering jargon using terms like 'support' and 'illegal' in a different way than normal contexts, as illustrated in Figure 19 and 20. To keep Susie credible, this metric has been marked as experimental and users first need to confirm they understand this (see Figure 15).

The sustainability check in the Governance section can also be inaccurate at times. For example, *Electricity maps* is a repository that maps energy usage all over the world [4]. This repository is built for analyzing energy usage and has a sustainable purpose. However, since it is not very explicitly mentioned in the terms that Susie searches for, it is not detected as addressing sustainability. The other way around can also happen. It is widely known that most Machine Learning and AI projects consume a lot of energy. However, this can only be detected when running the project. Susie only analyses the energy consumption of the programming languages and the amount of failing workflow runs. For example, *Tesseract* is a repository consisting of a machine learning project [13]. However, Susie classifies it as sustainable in terms of energy efficiency, as it mainly consists of C++ and does not have many failing pipelines. Actually, energy efficiency is the only metric that this repository performs well on, as it has a negative sentiment overall, the language is not very inclusive and there is no code of conduct.

Overall, we see that Susie can be a powerful tool to analyze repositories on certain types of sustainability and can provide developers with insights about their projects. However, it is very important to place the analysis in the context of the project and to be critical of Susie's output. Therefore, we have ensured that any user can be critical and needs to be cautious by adding many explanations, sources and warnings throughout the dashboard.

5 Social impact

To enable other developers to contribute to Susie, we have made the repository open-source. Issue templates and pull request templates are in place to ensure consistency is maintained, e.g. issues need to be clearly defined and are by default labelled with *Feature Request* or *Bug*. Next to that, a guide is added to the repository, detailing how to contribute, along with a code of conduct that we expect developers to follow. In order to make more developers aware of sustainability in software engineering and possibly help make this transition, Susie's website and open-source community needed some promotion. Therefore, we have posted on a few Reddit pages concerning themselves with open source projects, sustainable development and other similar topics. The promotion has already led to many upvotes, multiple GitHub stars and someone adding an issue to the repository. Hopefully, Susie keeps growing and can make a genuine impact.

6 Conclusion

All things considered, Susie is a powerful tool for analyzing the sustainability of public GitHub repositories, with a focus on social, individual, technical, economic, and environmental sustainability. By using a wide range of metrics (i.e. programming languages, inclusive language, governance, workflows, contributors, issue sentiment), Susie generates detailed audits that enable developers and other stakeholders to make informed decisions about the sustainability of the software they use and/or contribute to. With also the addition of guides on the website, Susie promotes sustainable software engineering practices and encourages developers to adopt more sustainable habits in their future projects. To fully incorporate the sustainability movement in the community, Susie has become an open-source project that welcomes contributions from any developer, such that we can create and speed up this journey together. Make sure to join at `https://github.com/philippedeb/susie`!

7 Future Work

Due to the short development span of the project, not all desired features could be implemented. However, with Susie being open-source, there truly is no limit for future work. Any cool idea, such as PDF audit generation, can be added. We have already created some issues in the repository, such that any (other) developer could implement amazing new features for Susie. Below, some of these ideas are described in more detail.

7.1 GitHub Token

When making requests to the GitHub REST API without a token, they are considered anonymous requests. This means that users will quickly run into the rate limit (number of requests that can be made per hour) that GitHub enforces for unauthenticated users if multiple analyses are executed subsequently. As there was no budget for a paid token or time to implement the option to enter your own token, Susie uses tokenless requests. For this prototype version, it is acceptable. However, if the project would be used on a larger scale, introducing tokens would be a necessity for high-frequency users.

7.2 Dockerfile Analysis

Many projects are deployed using Dockerfiles. Several aspects of these files can be analysed for sustainability. First, the image size plays a large role in the amount of storage space required and the time it takes to transfer it. Data storage and transmission can cost quite a lot of energy, so by optimizing the size of the image and reducing unnecessary dependencies, the environmental impact of a Dockerfile can be reduced by proper resource utilization. By using techniques such as caching and layering, the image build time can also be reduced and have a positive impact on the environment.

This feature was not implemented yet, since none of the team members has significant experience with Docker. Therefore, it was decided to focus on other features. It would still be a desirable feature to have for Susie and is thus an open issue in the GitHub repository.

7.3 Geolocation Analysis

Using geolocation data from GitHub comments can provide an indication of the contributor diversity working on a repository. Having different experiences and perspectives is important, as it can lead to a more inclusive product. It would also be a great indicator of the spread and awareness of a repository.

Relying solely on the geolocation data of GitHub commits might be an unreliable way of determining contributor diversity, and more research is needed to incorporate it in the GitHub data analysis. However, with the right methods and motivations, it could be a good metric to show on Susie and should be considered.

7.4 Carbon Calculator for Websites

Within the Governance metric, Susie checks if a repository has a website, as illustrated in Figure 11. When detected, it would be a remarkably insightful metric to calculate the amount of carbon the website emits using renowned third-party tools. For example, `www.websitecarbon.com` has an API and npm package that can provide this carefully curated carbon data [3]. As this service needs a key to operate, it was difficult to request and implement this feature during the short development span of Susie.

References

[1] alpacaaa. *Zero Bullshit Haskell*. https://github.com/alpacaaa/zero-bullshit-haskell. 2019.

[2] Luis Cruz. *Sustainable Software Engineering*. https://luiscruz.github.io/course_sustainableSE/2022/. Accessed: 29-03-2023.

[3] Ricardo Dantas. *website-carbon-calculator*. 2021. URL: https://www.npmjs.com/package/website-carbon-calculator.

[4] Electricity Maps. *Electricity Maps*. https://github.com/electricitymaps/electricitymaps-contrib. 2022.

[5] *Environmental Keywords - Find SEO Google AdWords Key Words for Your Website*. Dec. 29, 2021. URL: https://www.wordstream.com/popular-keywords/environmental-keywords (visited on 03/29/2023).

[6] Academy Software Foundation. *Inclusive language in technology*. Feb. 2021. URL: https://www.aswf.io/blog/inclusive-language/.

[7] *Glossary of sustainability*. URL: https://sustainable.org.nz/learn/tools-resources/glossary-of-sustainability/ (visited on 03/29/2023).

[8] Humbedooh. *Codebase Development Resilience*. https://humbedooh.com/Chapter%203,%20part%20one_%20Codebase%20development%20resilience.pdf. n.d.

[9] Nomic AI. *GPT4ALL*. https://github.com/nomic-ai/gpt4all. 2023.

[10] Rui Pereira et al. "Energy Efficiency across Programming Languages: How Do Energy, Time, and Memory Relate?" In: *Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering*. SLE 2017. Vancouver, BC, Canada: Association for Computing Machinery, 2017, pp. 256–267. ISBN: 9781450355254. DOI: 10.1145/3136014.3136031. URL: https://doi.org/10.1145/3136014.3136031.

[11] *Sentiment*. URL: https://www.npmjs.com/package/sentiment.

[12] Sustainable Review. *Sustainability 101: 35 terms and definitions you need to know*. Feb. 6, 2023. URL: https://sustainablereview.com/sustainability-101-terms-and-definitions/ (visited on 03/29/2023).

[13] Tesseract. *Tesseract Open Source OCR Engine*. https://github.com/tesseract-ocr/tesseract. 2016.

[14] webmech. *bad-words*. 2021. URL: https://www.npmjs.com/package/bad-words.

A Visual overview of Susie

This appendix provides a visual overview of the website. The figures included in this section show the various parts of the website, such that readers can understand the layout and functionality of the tool more easily and can provide a more concrete sense of how Susie works in practice. For the landing page, please refer to Figure 1. The guides page can be seen in Figure 2.

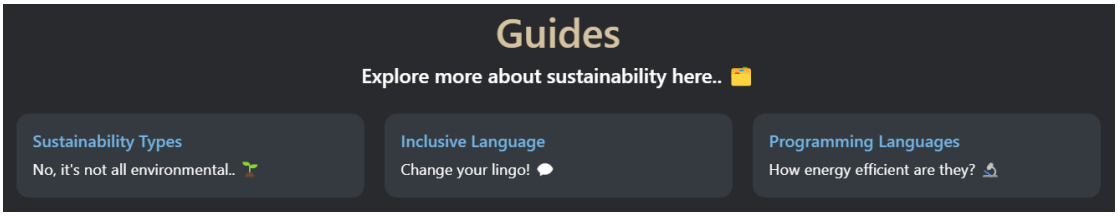


Figure 2: Guides section of Susie - to learn more about sustainable software development.

As an example for analysing a GitHub repository, a popular Python library named `Streamlit`[1] is used to demonstrate Susie below. Please read the captions in order to have an optimal storyline. Feel free to also check it out yourself, you can even analyze Susie with Susie!

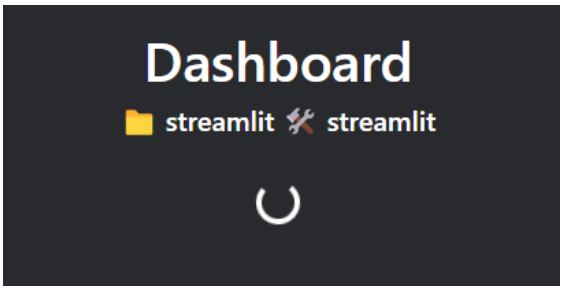


Figure 3: When Susie is analysing a repository, an animated loading icon is displayed.

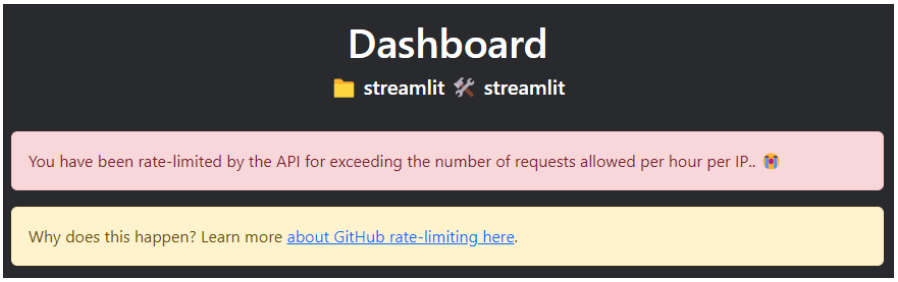


Figure 4: When Susie fails to analyse a repository due to various reasons, the user gets an alert with the reason why Susie failed. In this example, the user has analysed too many repositories in a short timeframe and the GitHub API has rate-limited the user, making Susie temporarily unusable.

[1] `https://www.github.com/streamlit/streamlit`

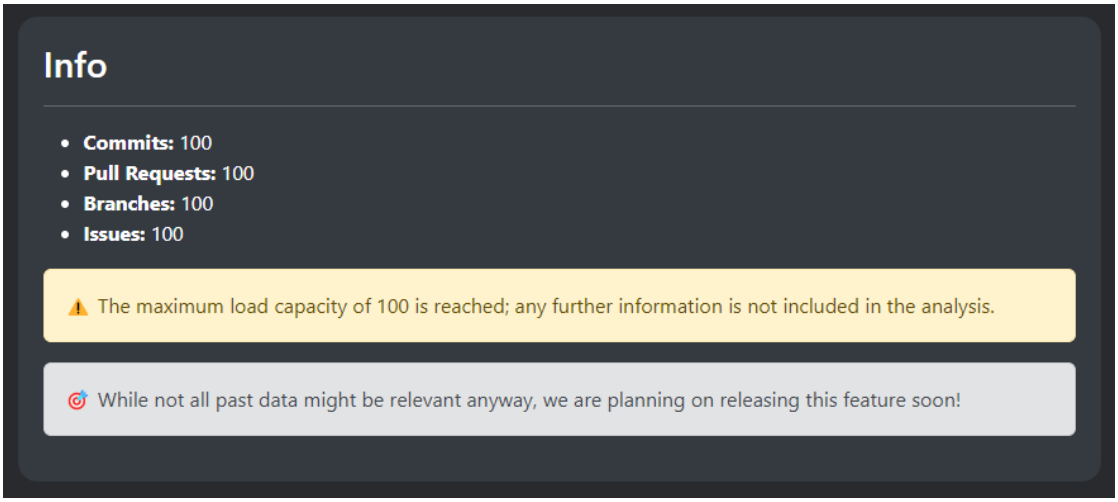


Figure 5: When the repository is analysed, Susie displays several sections in a dashboard. The first one is *Info*, which gives a brief overview of the repository. Additionally, it shows any warnings regarding fetching the data.

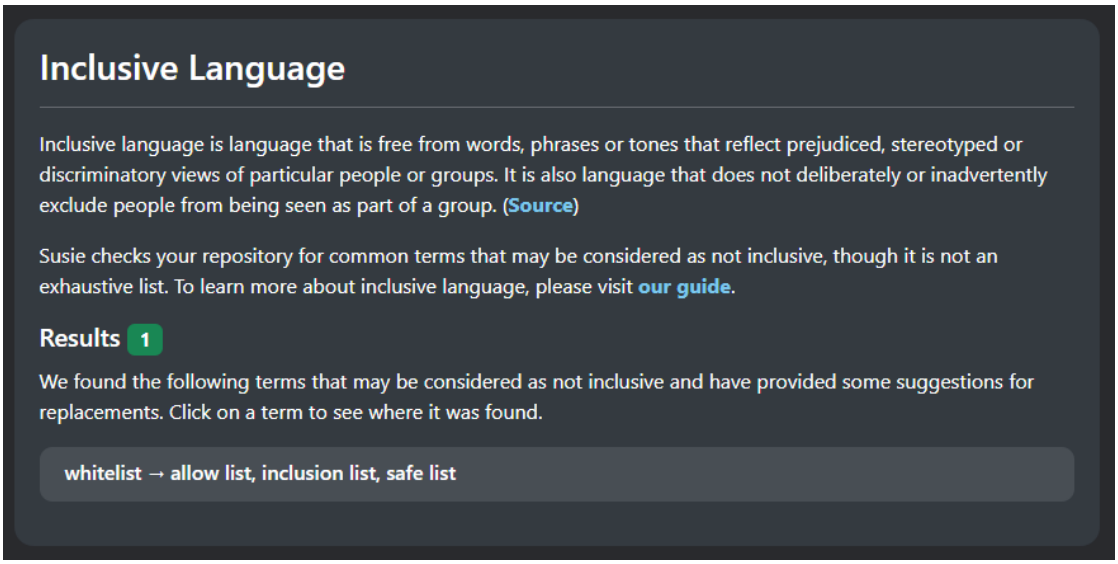


Figure 6: The *Inclusive Language* section in the dashboard. Any suggestion can be expanded by the means of a dropdown component and will list the origins of the respective suggestion, see Figure 7.

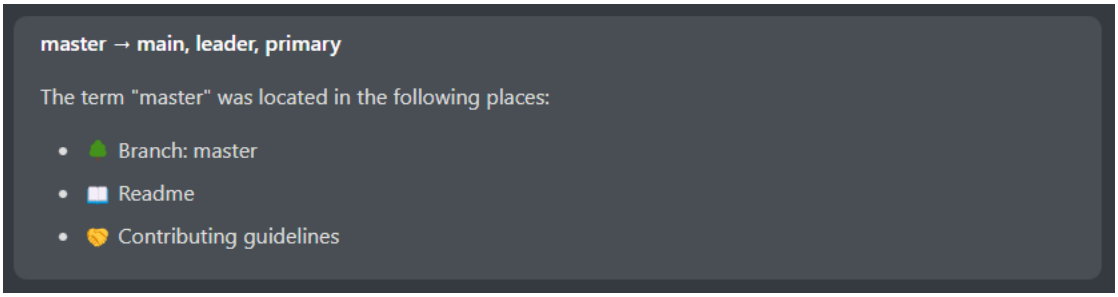


Figure 7: In the *Inclusive Language* section, any suggestion can be expanded to view the origins of the respective suggestion. To keep it concise, the origins' data types are captured and mapped to simpler messages, e.g. `README.md` is not completely listed, only a simple pointer called 'Readme'. Only commits and issues are listed more elaborately.

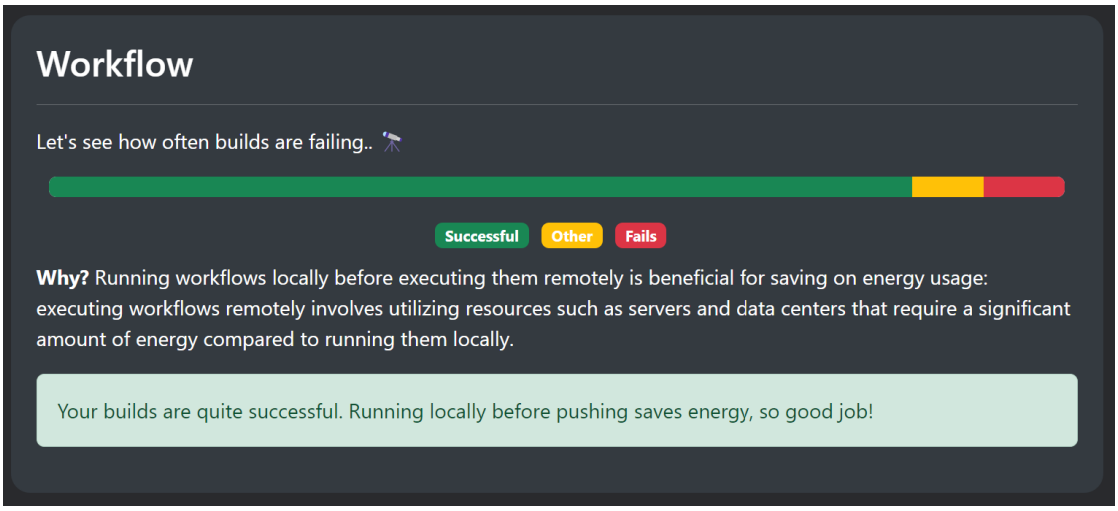


Figure 8: The *Workflow* section in the dashboard. If no data is found, the bar is grey and has no legend.

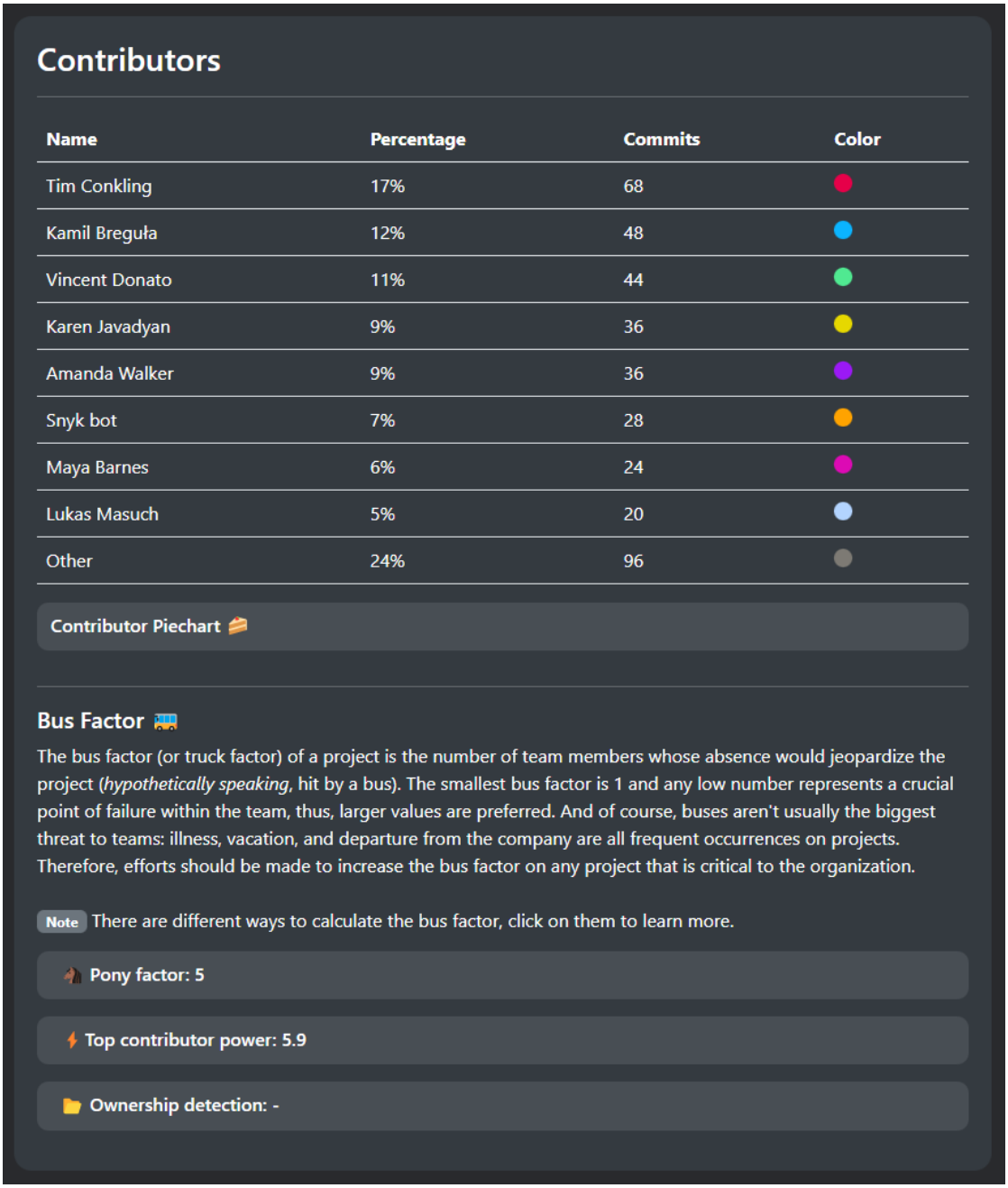


Figure 9: The *Contributors* section in the dashboard. Provides insight into the distribution of work done by all contributors. To keep it concise, the piechart and extra details are hidden in dropdowns, see Figure 10.

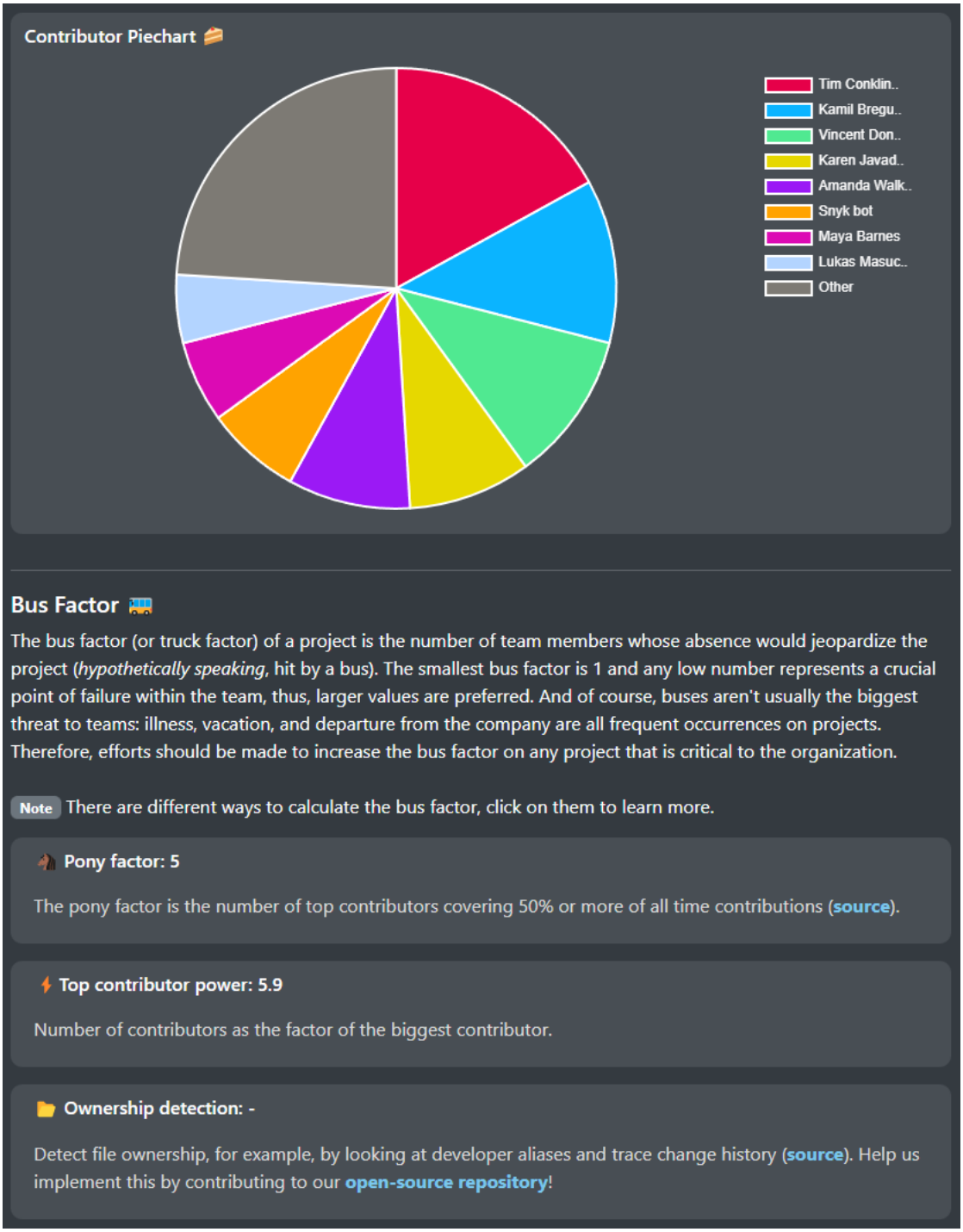


Figure 10: Extra information can be retrieved by expanding the dropdowns in the *Contributors* section. For example, how the bus/truck factor is (or could be) calculated by various metrics.

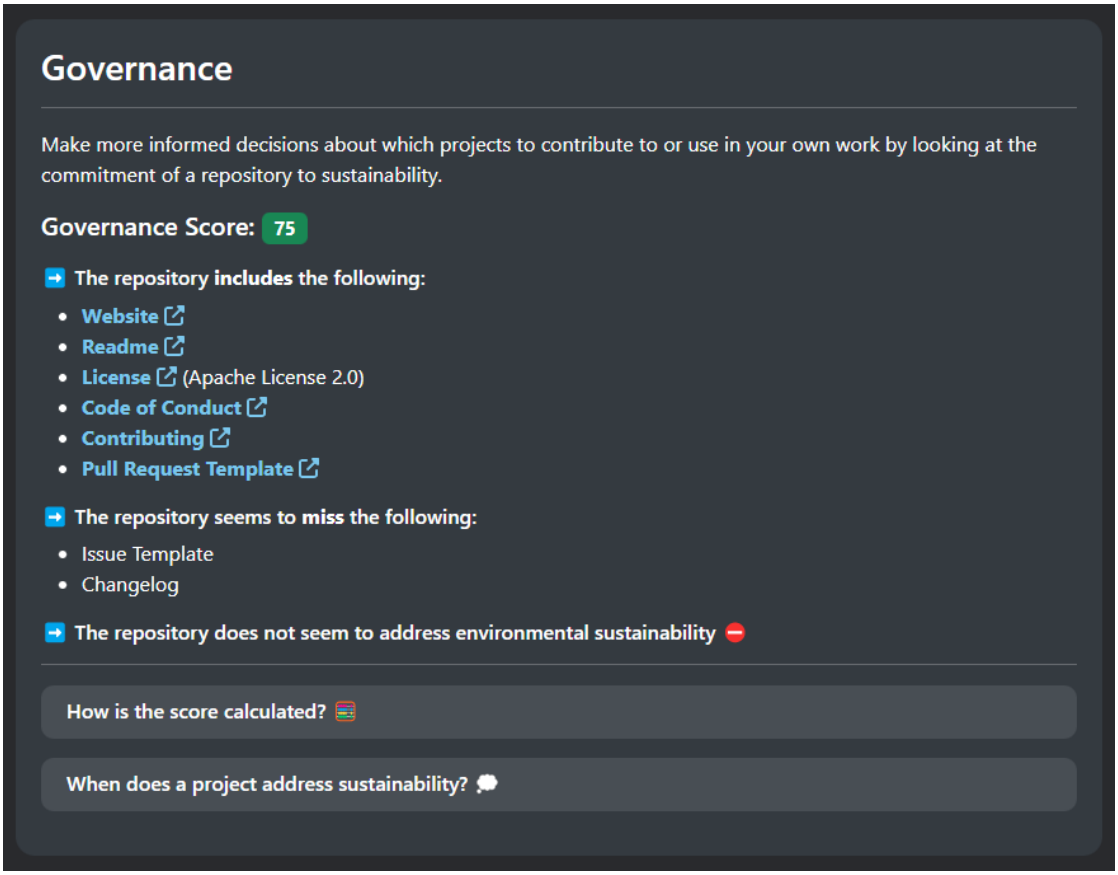


Figure 11: The *Governance* section in the dashboard. Provides a score based on what is included and excluded, in addition to direct links for easy inspection. Furthermore, any details about the component are provided in the dropdown components at the bottom. For example, all the terms listed in Section 2.1.3 are mentioned here as well, see Figure 12.

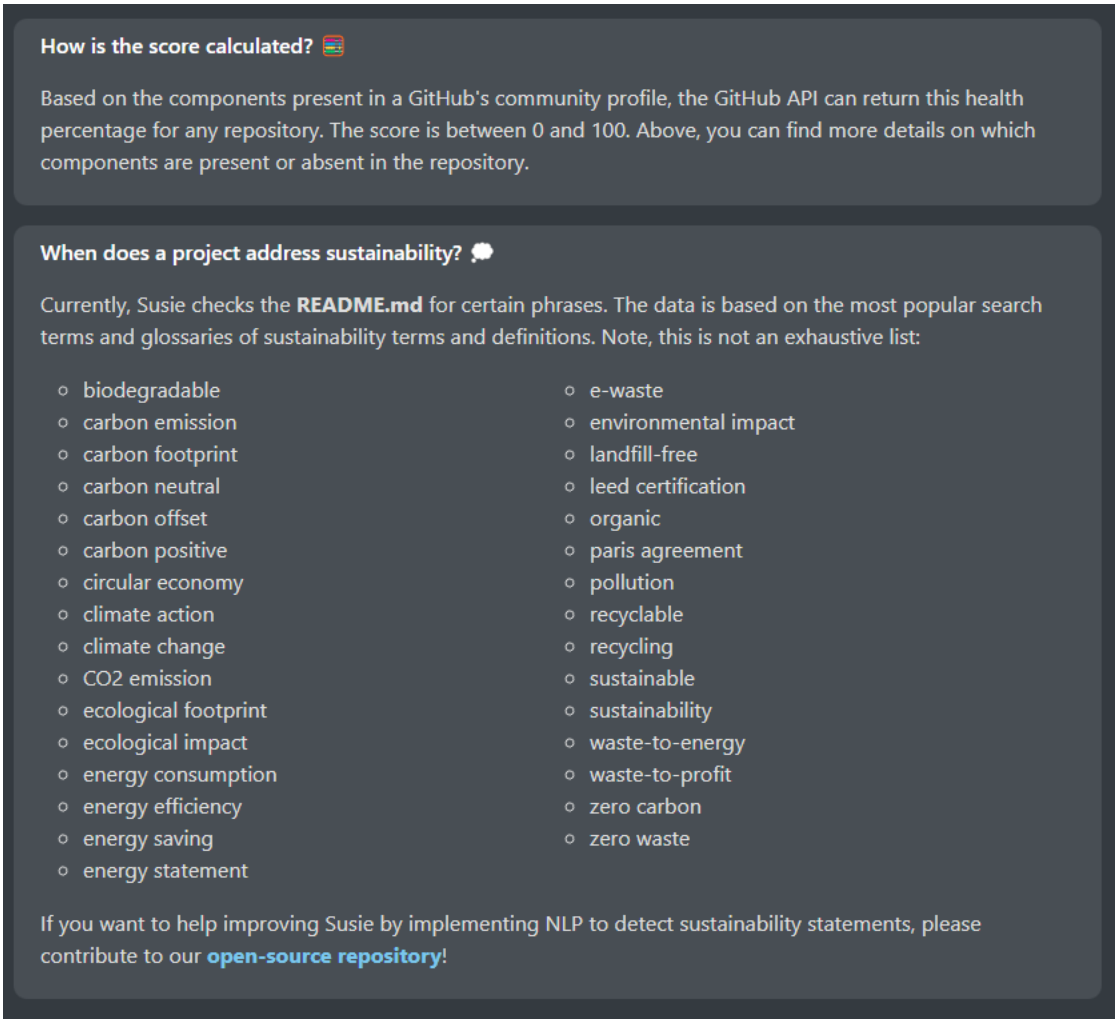


Figure 12: Details behind the *Governance* section are provided in dropdown components at the bottom of the metric. The score and sustainability terms are explained thoroughly here.

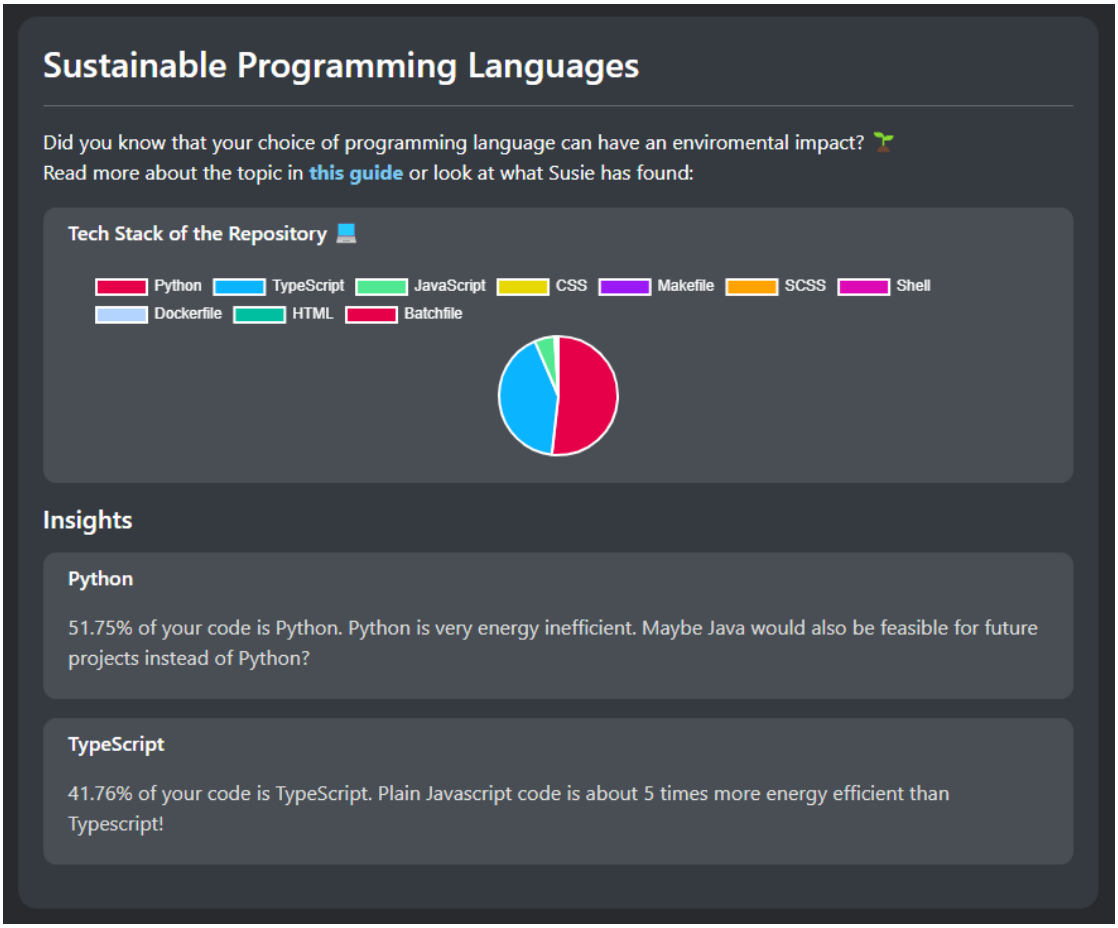


Figure 13: The *Sustainable Programming Languages* section in the dashboard. An animated and responsive piechart using `Chart.js` shows the tech stack and Susie provides insights for the most prominent languages used in the GitHub repository. All inner sections can be minimized by clicking on the bold titles, in case the user finds the number of details becoming too chaotic when browsing through the Susie dashboard.

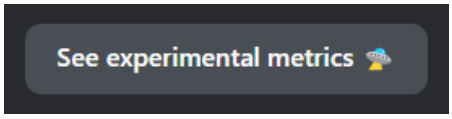


Figure 14: At the bottom of the dashboard, users can decide whether they want to see more experimental metrics of Susie. First, they have to confirm and read a note, as illustrated in Figure 15.

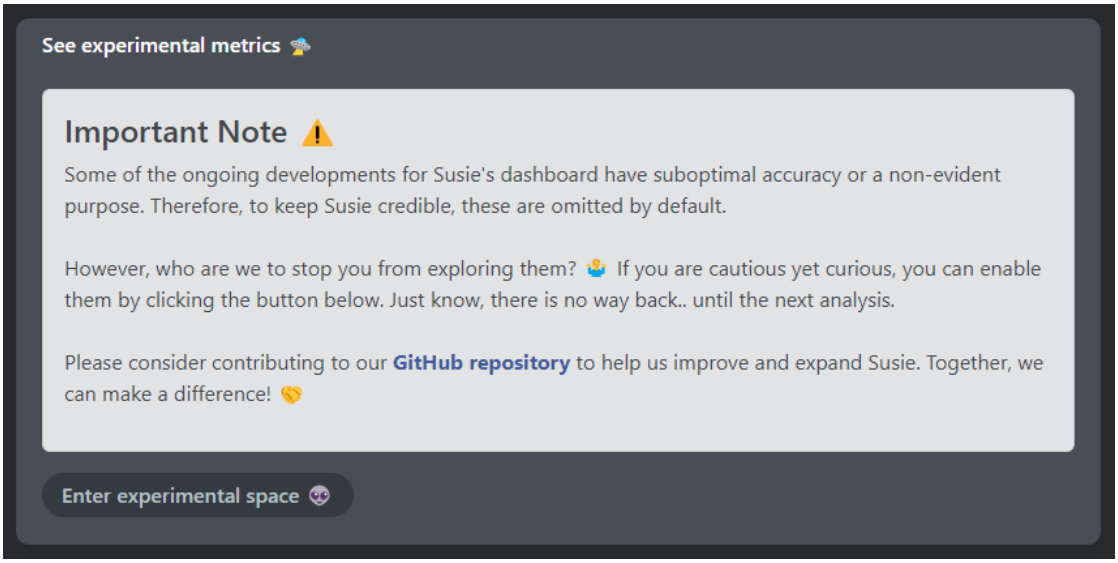


Figure 15: Users are warned about the accuracy or evidence behind the experimental metrics before they continue, in order to keep Susie credible yet progressing. After entering the experimental mode, the menu expands (instead of Figure 17) and new metrics are displayed, e.g. Figure 16.

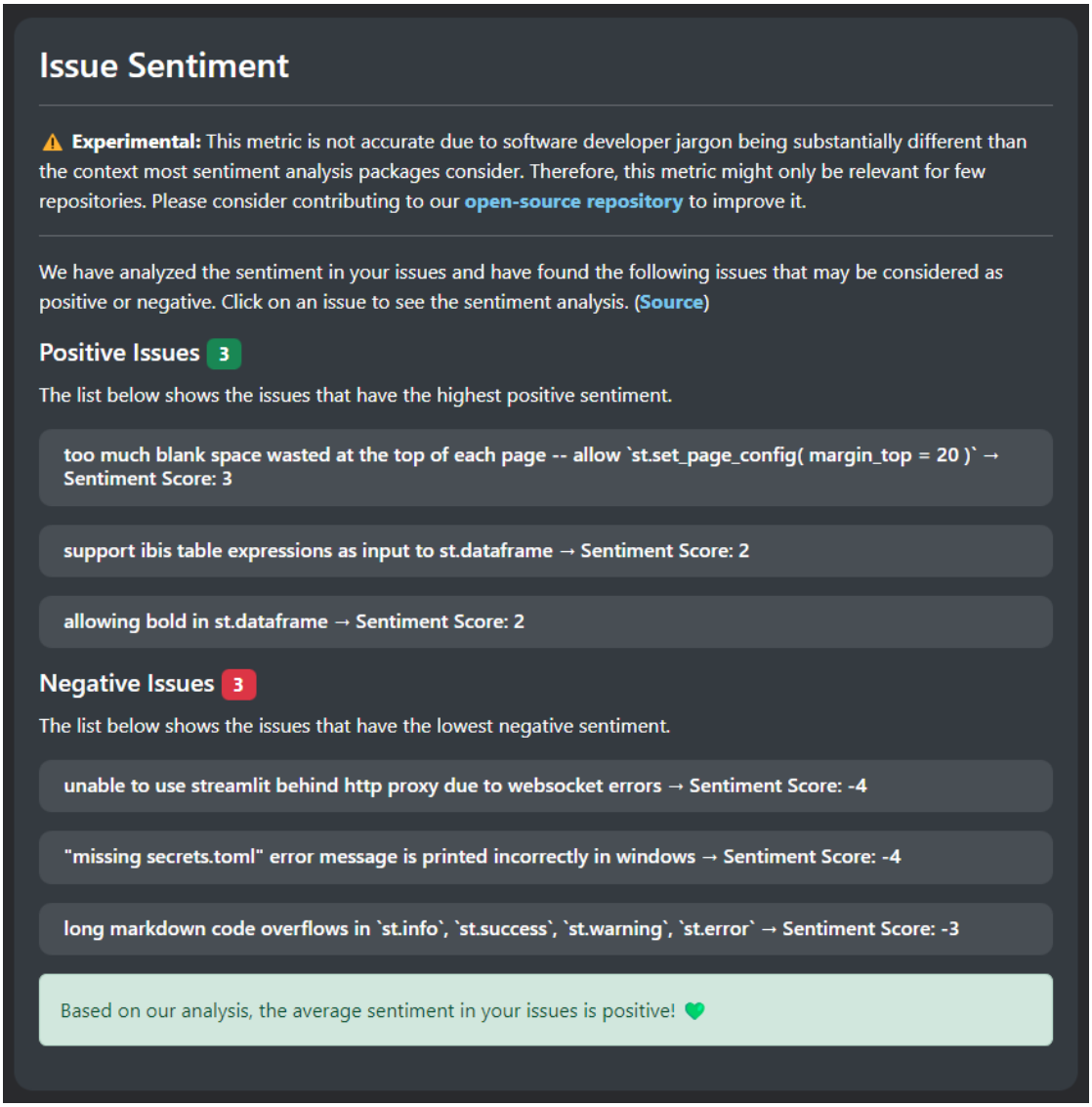


Figure 16: The *Issue Sentiment* section in the dashboard. This metric is marked as experimental, due to its suboptimal accuracy (see Validation, Section 4).

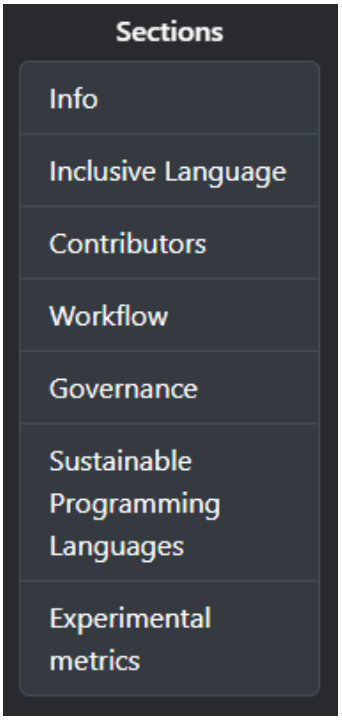


Figure 17: With so many sections in the dashboard, Susie also provides a floating sidebar on non-mobile devices to smoothly browse through the dashboard (animated scroll).

B Validation Samples

This appendix provides additional validation samples for the metrics used in Susie. Please refer to Section 4 for more information.

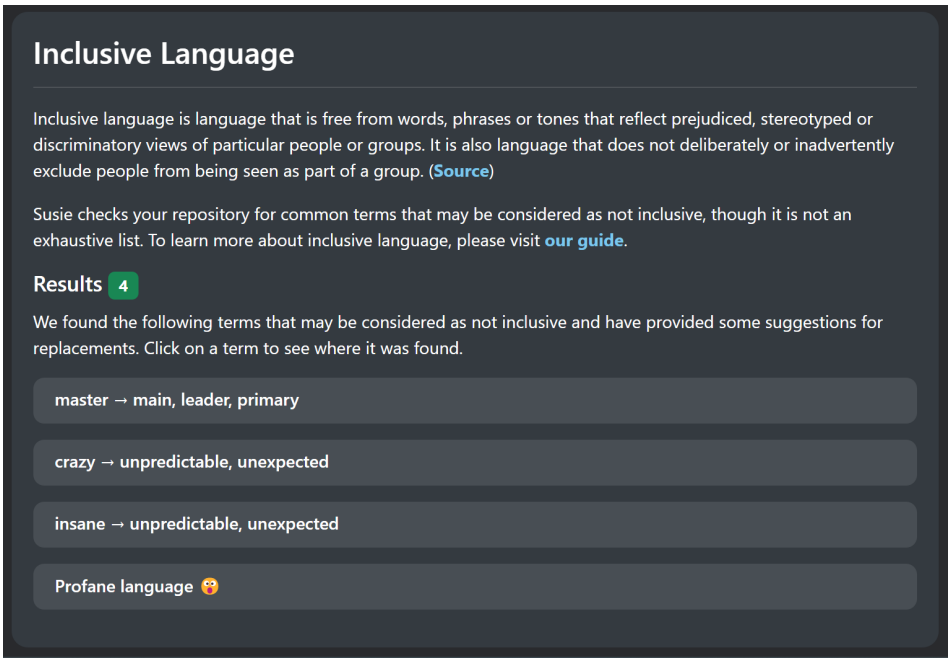


Figure 18: Validation sample for the inclusive language metric.

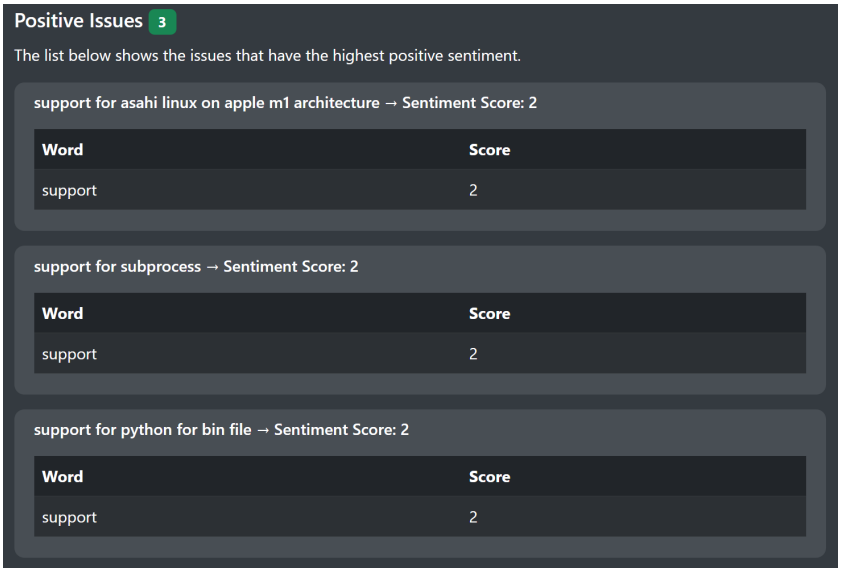


Figure 19: Validation sample for the issue sentiment metric (in this case, positive sentiment).

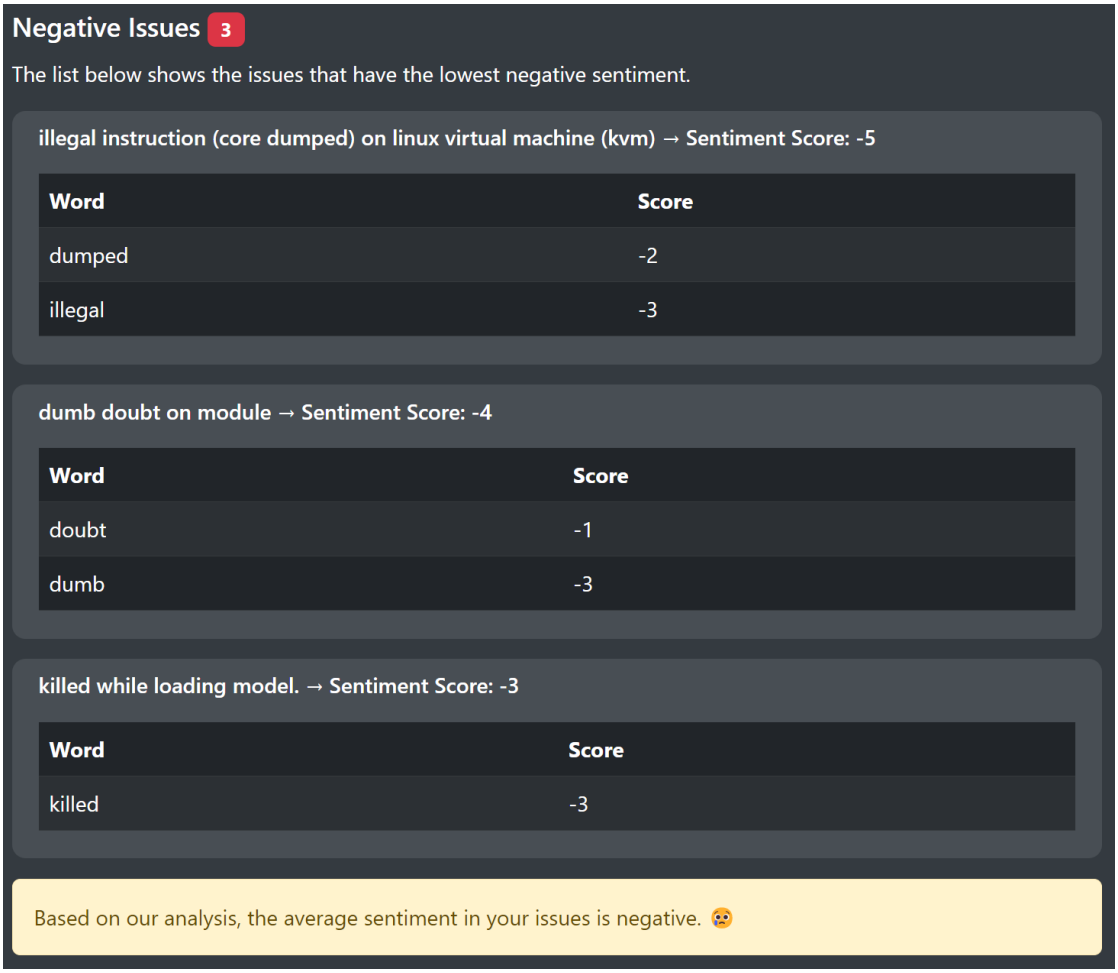


Figure 20: Validation sample for the issue sentiment metric (in this case, negative sentiment).