Aspirations and Practice of ML Model Documentation: Moving the Needle with Nudging and Traceability

Avinash Bhat*
McGill University
Canada

Austin Coursey*
Vanderbilt University
United States

Grace Hu
McGill University
Canada

Sixian Li
McGill University
Canada

Nadia Nahar
Carnegie Mellon University
United States

Shurui Zhou
University of Toronto
Canada

Christian Kästner
Carnegie Mellon University
United States

Jin L.C. Guo
McGill University
Canada

ABSTRACT

The documentation practice for machine-learned (ML) models often falls short of established practices for traditional software, which impedes model accountability and inadvertently abets inappropriate or misuse of models. Recently, model cards, a proposal for model documentation, have attracted notable attention, but their impact on the actual practice is unclear. In this work, we systematically study the model documentation in the field and investigate how to encourage more responsible and accountable documentation practice. Our analysis of publicly available model cards reveals a substantial gap between the proposal and the practice. We then design a tool named DocML aiming to (1) nudge the data scientists to comply with the model cards proposal during the model development, especially the sections related to ethics, and (2) assess and manage the documentation quality. A lab study reveals the benefit of our tool towards long-term documentation quality and accountability.

CCS CONCEPTS

• **Human-centered computing → Interaction techniques**; • **Software and its engineering → Application specific development environments**.

ACM Reference Format:
Avinash Bhat, Austin Coursey, Grace Hu, Sixian Li, Nadia Nahar, Shurui Zhou, Christian Kästner, and Jin L.C. Guo. 2023. Aspirations and Practice of ML Model Documentation: Moving the Needle with Nudging and Traceability. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3544548.3581518

*Both authors contributed equally to this research.

CHI '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9421-5/23/04…$15.00
https://doi.org/10.1145/3544548.3581518

1 INTRODUCTION

Documentation serves as the primary resource to understand and evaluate reusable software components when adopting them in developing applications [1]. Machine-learned (ML) models increasingly are integrated as components into software systems and would benefit from similar documentation [3, 36, 48]. Stakeholders, including data scientists, AI engineers, domain experts, and software engineers, resort to the documentation to answer questions such as what use cases are supported, what performance to expect, and what ethical and safety impacts to consider once the model is deployed in applications at scale. Nevertheless, ML models shared as pretrained models or services are often poorly documented. Still they are reused in many applications, sometimes in applications for which they were not designed. Serious issues related to misuse of ML models have been observed in various applications, notably in face recognition and tracking [10], recruitment [17], and criminal risk assessment [19], leading to broader concerns about their impact on social justice.

As a reaction to observed problems in ML model reuse and accountability, significant efforts towards documenting models [4, 45] and data [24] have been proposed. This line of work has attracted considerable attention – for example, the paper on *model cards* published at FAccT 2019 [45] is heavily cited, and the popular model hosting site HuggingFace has adopted the term *model card* in their user interface and guides their users to provide documentation [21]. Yet, it is largely unknown how these proposals have impacted the practice of documenting ML models and datasets.

In this work, we systematically study ML model documentation in the field and investigate how to encourage more responsible and accountable model documentation practice. While past work has already shown often limited documentation during model development, such as few markdown cells in public notebooks and missing README files in notebook repositories on GitHub [51, 57], we focus on external documentation of reusable ML models and services. We start by investigating how the ML models and services made available are documented, in particular, how they meet the model cards template proposed by Mitchell et al. [45]. Our study reveals that despite adopting the model card terminology, most model development teams fail to provide meaningful and comprehensive documentation that can support scrutiny for model

adoption. Certain aspects of documentation are especially limited across different contexts of model development (i.e. open-source and proprietary), such as information regarding the data collection process, evaluation statistics explanation, and concrete ethical measurements.

Motivated by the observed low adoption rate of the model cards proposal and frequent poor documentation quality even when the proposal is followed, we explore how we could improve the adoption of model cards and encourage good documentation practices. To this end, we design and implement a documentation tool for data scientists, named DocML, that supports creating and updating user-oriented documentation during model development in the computational notebook environment. A user study with 16 participants demonstrates that, when DocML was presented, data scientists adopted documentation approaches that benefit accessing and managing model documentation quality. They also showed more deliberate consideration of model development context and ethical concerns.

Our work makes the following contributions to understanding and supporting ML model documentation practice:

(1) Results from our empirical study, that delineate the current practice of public model cards and highlight a clear gap between the information needed for the model users and information provided by the model developers;
(2) A rubric for evaluating ML documentation, developed and used in our study based on model cards proposal, which can be adopted by model developers and users as a documentation guideline or quality assessment tool;
(3) A JupyterLab extension, DocML to support data scientists to write, inspect and maintain model cards during the model development process, evaluated in a user study.

The artifacts created in this study including the rubric, list of assessed model cards, user study design, and DocML source code, are shared as supplementary materials alongside the paper in the ACM Digital Library to support future investigation on improving ML documentation.

2 RELATED WORK

In this section, we discuss how our work is situated in previous literature on software documentation, machine learning documentation, and tool support for data scientists.

2.1 Software Documentation

Documentation plays a key role in various software qualities, such as usability and maintainability [60]. The primary information about the objectives, design, and usage of the software is recorded in different types of documentation. The study of software documentation concentrates mostly on the aspects of documentation property and quality [2, 5, 52], documentation search and discovery [62, 63], content augmentation [56, 64], and documentation creation support [31, 32, 46]. Among them, our work is most relevant to the previous inquiry on documentation quality and interactive documentation creation support.

Through a survey study with 323 software professionals at IBM, Uddin and Robillard identified ten common API documentation problems that manifested in practice [65]. Among those problems, *incompleteness* and *ambiguity* were considered the most frequent problems that caused severe impacts. A recent study by Aghajani et al. examined documentation problems through a data-driven approach [2]. They developed a taxonomy of documentation issues by analyzing the documentation-related discussion developer mailing lists, Stack Overflow discussions, issue repositories, and pull requests. *Completeness* and *up-to-dateness* are frequently mentioned. Together with *correctness*, they constitute the category of issues concerning documentation content. At the same time, issues beyond documentation content are extremely common and have profound implications for the documentation writers, readers, and maintainers, such as how the content of the documentation is written and organized (e.g., documentation usability and maintenance), documentation process (e.g., traceability and contribution), and documentation tool (e.g., bugs, supports, and improper tool usage). This taxonomy illustrates the complexity of documentation concerns and calls for a consideration of documentation within the context of software development.

A large body of work on supporting documentation creation aims to automate content generation. Examples include generating progress-related documentation such as commit messages [15] and summarizing method [44], files [46], or even the whole project [32]. Such work normally relies on heuristic or machine learning methods to extract or synthesize content from the input artifacts and inevitably introduces both errors and biases. Since our work emphasizes documentation quality, a more interactive approach with which the documentation writer has full control over the content being created is more relevant. The work by Head et al. is an example of focusing on the interaction aspect for tutorial writers [31]. We adopt a similar approach. Motivated by the empirical observations of model documentation quality, we seek to address the needs from data scientists during model development and documentation through the interaction design.

2.2 Documentation for Machine Learning

The interest in ML documentation mostly concerns data and model documentation [8], proposing what content to include in such documentation. On the data side, work on *Dataset Nutrition Labels* [33] and *Datasheets* [24] propose standards for documenting information related to the data, such as provenance, statistics, and accountable parties. Using software modeling techniques, the work of *DescribeML* proposes to describe the dataset structure, provenance, and social concerns (e.g. biases, potential harm, and privacy) through a domain specific language [25]. On the model side, *model cards* [45] and *fact sheets* [4] propose standards for model documentation. Particularly, the work on model cards proposed by Mitchell et al. [45] has gathered substantial interest from both academia and industry. It aims to standardize ML model documentation; it suggests that the model cards should record information beyond performance characteristics, including intended and out-of-scope use cases, potential pitfalls, and ethical considerations. Several companies such as Google, Nvidia, and Salesforce have adopted model cards for some of their public models. Hugging Face, a open-source ML model hosting platform, also encourages its users to adopt model cards when sharing their models. Our work provides a more critical view of the current adoption of model cards. We set off to

understand the impact of the model cards proposal on the quality of model documentation.

Previous work on the ML documentation process is relatively scarce. For *fact sheets*, Richards et al. [55] discuss that an interactive process with stakeholders is needed to define what information should be contained in the documentation in the first place. The proposed methodology describes how stakeholders can instrument their documentation generation at each stage of the AI development life cycle by asking concrete questions relevant to that stage. While the outcome of this process might resemble the model cards, it provides more support for planning the documentation effort and collaboration between different roles within the organization towards AI documentation. A more recent work built on the concept of model cards delves deep into how the content of model cards can support non-experts in making decisions related to the model [16]. Compared to the standard model cards proposed by Mitchell et al. [45], this work and its notion of *"interactive model cards"* shift the focus to the model users and provide more probing and assessing support for them to better understand the risks and ethical consequences related to model adoption.

Despite the intense interest in ML documentation, in practice, the effort and therefore the quality of documentation still fall short. In a recent study with 45 practitioners from 28 organizations, documentation is identified as one of the biggest challenges when building and deploying ML systems into production [48]. Similar to traditional software, incomplete and outdated documentation is a major concern. On the data side, the existing data documentation for public datasets is "never sufficient for model teams to understand the data" [48]. On the model side, missing documentation causes hidden assumptions and losing knowledge on key decisions about the models being developed. Our work contributes to filling the wide gap between aspiration and practice for ML documentation. In particular, as a starting point, we aim to nudge the data scientists to consider various aspects of model cards during model development and adopt better practices towards model documentation.

2.3 Tool Support for Data Scientists

As part of the ML development team, data scientists fulfill a decisive role in shaping the machine learning pipeline [47]. From available public or internal data sets, data scientists perform complex data wrangling to understand and transform the data into usable formats for ML models. They also experiment with different model architectures and hyper-parameter settings to improve the model performance on important metrics. The entire process is iterative and often performed on computational notebooks, such as Jupyter notebooks and Google Colab [71]. Computational notebooks are effective when used as a scratch pad to quickly test ideas or to create a narrative computational storyline, but at the same time, they are reported to suffer from many problems, such as missing version control, unpredictable executing orders, and ill support for managing dependencies and debugging, creating significant barriers for data scientists to adopt best engineering practices and to reliably develop ML models [12].

Existing work supporting data scientists' workflow mainly focuses on the notebook environment. The effort includes but is not limited to synthesizing data wrangling code [20], visualizing and comparing alternative paths in the ML pipeline [67, 69], and supporting cleaning of exploratory code [29]. In terms of documentation, Yang et al. [70] proposed a documentation tool WrangleDOC to automatically summarize the data wrangling code in the notebook through program synthesis. The generated summary consists of data input, output, and transformation that can assist the data scientists in understanding and verifying the early steps of the ML pipeline. Along similar lines, Wang et al. [68] proposed a tool called Themisto that combines deep-learning and information retrieval approaches to generate documentation for code cells in the notebook. Themisto also prompts the data scientists to add documentation for the code cells with output. Compared with those works, our tool has a distinct objective. The target users of WrangleDOC and Themisto are data scientists themselves. Those tools aim to support documenting the notebooks for data scientists to develop and reuse notebooks. Therefore, the resulting documentation can be filled with developmental details. Maffey et al. [43] propose a domain-specific language for model evaluations that require embedding in the development process and that can automatically generate reports as documentation including many different facets of model quality. The tool proposed in our work, however, focuses on encouraging data scientists to consider ethical aspects of their model development and to follow the documentation standard to create documentation for various stakeholders who need to reason about whether the model properties in different use cases.

3 UNDERSTANDING MODEL CARD PRACTICE

The number of ML models being published and reused is increasing at an astounding speed. *Hugging Face*, one of the popular platforms for sharing and hosting reusable models, is used by more than 5,000 organizations and currently hosts more than 19 thousand machine-learned models [23]. The top-ranked model on Hugging Face named *BERT base model (uncased)*[1] is downloaded more than 22M times per month (accessed on August 2022). Many organizations also offer proprietary models for a wide range of tasks through public APIs, from BigTech companies such as Google and AWS to many startups.

The technical steps for reusing models and incorporating them as components into applications for various predictive tasks are easy, typically by downloading the trained model binary or calling a REST API. However, understanding the scope and quality of a model is often not obvious. Incomplete documentation of ML models can cause serious trouble for potential model adopters to properly set up the models within their own application. More importantly, without information about the model development process and their impact on performance and ethics in the application domain, the models might be misused or used without proper care, therefore causing various harm to the end-users [7, 10].

To understand the current practice in documenting reusable models and the gap between recommendations and practices, we conduct an empirical study on model documentation. We start with collecting a dataset of models and corresponding documentation that explicitly or implicitly indicates the adoption of the proposal of *model cards*. We focus our study on model cards because this format

[1] https://huggingface.co/bert-base-uncased

has had a considerable impact in both academia and industry [45] (more in Section 2.2). We then examine the collected model cards using a rubric we created based on the original proposal for model cards [45]. Our analysis is entirely manual and mixes qualitative and quantitative aspects to ensure the reliability of our evaluation. We developed and validated our rubric iteratively and release it publicly as a potential foundation for documentation guidelines or quality assessment tools. Finally, we discuss the implications of the model documentation quality evaluation result.

3.1 Background: The Model Cards Proposal

We first provide more details about the work of model cards. Before this work, ML models were mainly compared based on model performance, measured mostly by metrics on the whole test dataset, such as precision, recall, and f-measures. The proposal of "model cards," suggests that the model comparisons should consider the ethical axes especially when the model is going to be adopted in applications that have a serious impact on people's lives. It further advocates that the model should be evaluated on the performance of subgroups divided by culture, demographic, other domain-relevant conditions, and their intersections. Overall, the model cards proposal suggests including nine sections for the model documentation to encourage more responsible and accountable practice during model development. The sections are:

- **Model Details**, lists basic information about the model, such as model release date, its version, the type of the model, license, responsible parties, and how to contact them;
- **Intended Use**, describes the primary use cases and users that the model serves and the use cases that are out-of-scope but easily confused with or highly related to;
- **Factors,** records how the demographic or phenotypic groups, as well as instrumental and environmental factors, impact the model performance;
- **Metrics**, covers the measurement of model performance, including how those measurements are calculated;
- **Evaluation Data**, describes the details of the datasets used to quantitatively evaluate the model performance. It should also include the justification of dataset selection and any prepossessing procedure followed;
- **Training Data**, describes the details of the dataset used for training the model. When the information cannot be disclosed, it should provide basic information such as the distribution over groups;
- **Quantitative Analyses**, illustrates how the model performs through disaggregated evaluation with respect to each factor identified and their intersections;
- **Ethical Considerations**, discusses the ethical considerations taken during the model development, such as if the model uses sensitive data, the foreseen risks and how they are mitigated, etc.;
- **Caveats and Recommendations**, lists additional concerns that are not covered in previous sections.

To illustrate how to adopt model cards, the original proposal included two concrete model cards documenting two publicly available models: one smiling detection model [42] and one toxicity detection model [37]. Interested readers can refer to the original model cards paper by Mitchell et al. [45] for full details.

3.2 Assessing Model Cards In the Field

3.2.1 Model Cards Collection. To understand how the model cards proposal is adopted in practice, we curated a dataset of model cards from three sources. We intentionally stratified our sample to cover mostly models that adopt the idea of model cards and to cover both commercial and research models. As a baseline for comparison of documentation practice, we also collected a set of model documentation that does not explicitly mention model cards. The final collection is summarized in Table 1. We describe the collection process from each source below.

Hugging Face Model Cards. We selected Hugging Face [23] as a source because of its large user base. They formally adopt the model cards proposal by providing documentation [22] and training materials [21] and show each model's README file under the label "Model Card" on the landing page for each hosted model. The content and structure of that README, however, are not checked when publishing models. We collected model cards from Hugging Face to observe how effective model card promotion can be. From its website, we collected all 370 models with more than 5,000 monthly downloads. We then randomly sampled 20 of these model cards from the top 100 monthly downloads, representing the most popular models on Hugging Face, and 30 from the rest of the 270 model cards, representing models with decent popularity.

GitHub Model Cards. GitHub is a popular platform to share ML models, along with the code to train and evaluate the model. Some authors have adopted the model cards proposal for documenting the model in their README files. Therefore, we included GitHub to analyze the practices of adopting model cards for open-source ML models. We used two search queries to identify candidate repositories. First, we used code search to identify repositories that used the Model Card Toolkit [26], an open-source Python Model Card API with the search query *"import model_card_toolkit"*. Second, we searched with the query *"model card"* in the README files among all repositories. We then manually validated all results, by removing any repositories that did not contain actual model cards, were included in the Hugging Face data, or were duplicates of another GitHub repository. After validation, we identified only a single repository using the model card toolkit and 23 repositories recognizably adopting the model cards proposal in their README. This process yielded a relatively complete set of model cards for ML models shared on GitHub.

Company Model Cards. As the third source of model cards, we searched for models offered as APIs from companies (from Big Tech to startups). To this end, we relied on Google search with keywords "model card" and "model card [company name]," using company names including Nvidia, Microsoft, Google, Facebook, OpenAI, DeepMind, and Amazon. We manually inspected the top results, discarding false positives, resulting in 28 model cards for commercial models. While the resulting set is not exhaustive, its size is comparable to the size of documentation sets from other sources.

Baseline (Non "Model Cards"). We further included a sample of ML models hosted on GitHub that do not claim to have followed

Table 1: Model cards collection that is used in our empirical study on documentation quality.

Source	Subcategory	# of Samples
Hugging Face	Top 100 Most Downloaded	20
	Top 101-370 Most Downloaded	30
GitHub Model Card	Model Card Toolkit	1
	"Model Card" in Project README	23
Companies	-	28
Baseline (Non "Model Card")	-	30
Total		132

the model cards proposal, representing the common unstructured model documentation practice. To identify relevant repositories, we searched GitHub for three common and popular machine learning tasks for which models are commonly shared and nontrivial reuse questions arise (including ethical questions): object detection, sentiment analysis, and face generation. Among the identified candidate repositories, we sampled 30 (10 for each task) that meet the following criteria: have a README, release their pre-trained ML models, but do not mention model cards.

3.2.2 Rubric Development. We realized in the early phases of the model cards assessment the difficulty in judging reliably how well certain aspects of a model are documented, for example, if the scope of a model is described accurately and clearly. Such judgment is highly subjective; we found low inter-rater reliability and found it challenging to define and describe levels of a measure. Hence, we converged on an approach that measures more reliably whether certain topics are covered in the documentation with concrete yes/no questions at the cost of capturing only the presence of information in the documentation but not its comprehensiveness or correctness. The resulting list of questions served as a rubric to judge how different aspects of the model cards proposal were documented and to compare model cards from different sources.

Concretely, starting from the description of each component in the original model cards paper, we converted each aspect to be covered in the recommended model card structure into a set of concrete yes/no questions. For example, for the aspect "primary intended usage" in the Intended Uses section of model cards, our rubric includes a question of *Does this model card (or equivalent model documentation) explain scenarios in which to use the model?* We developed those questions iteratively based on our own observations while inspecting documentation in our dataset. For example, we observed that certain aspects recommended by model cards are closely related, for example, the "Quantitative Analyses" and "Ethical Considerations." We merged those sections and added concrete questions to resolve potential ambiguity in interpreting their categories. The resulting questions for "Ethical Considerations" include: *Are ethical considerations discussed? Does the documentation discuss the used* ***process*** *for considering ethical issues with the model?* and *Do the documentation provide* ***concrete measurements*** *to support the discussed ethical considerations?* (Q20 - Q22 in Table 2). For each section, we also added the interpretation of potentially ambiguous terms in the questions, as well as examples of when to rate yes and no for corner cases.

The rubric was created iteratively and underwent three rounds of inter-rater reliability assessment. An initial rubric was designed and used by six individual raters, each evaluating the same ten model documentations (random selection of five from companies and five from Hugging Face). We then updated our rubric after investigating and resolving inconsistencies among the raters. We followed the same process during the second round on a new set of 15 model cards with an inter-rater reliability of 0.59 using Cohen's Kappa [14]. In the final round, we focused on three questions that yielded the most disagreement: questions about the target distribution of the model and the description of the training data (Q8, Q17, and Q19 in Table 2). After clarification and refining the rubric for those questions, we reached 0.73 inter-rater reliability using Cohen's Kappa for those questions using additional 15 model cards.

As part of the rubric, we instructed raters to only look at the primary documentation (model cards or README files), but not to follow links to papers or external data, unless the main documentation makes it clear what specific information can be found there (e.g. "for more graphs demonstrating the model's performance, see *link*"). This was an intentional choice to focus on the core documentation that a user might read rather than evaluate what information users could acquire by digging through academic papers or conducting their own experiments. This aligns with the model cards proposal that suggests collecting all important information in a compact description.

After establishing the reliability of the assessment rubric, one author manually rated all model cards in our dataset using the rubric. Note that we do not intend this rubric to give an overall score to a model card, but intend it to be used to determine whether the different aspects of information recommended by the model cards proposal are provided. The complete rubric is included in the supplementary material for future reuse and refinement by model developers and other model stakeholders.

3.2.3 Threats to Validity. As discussed, our rubric does not assess the completeness or correctness of information, but only whether a model card includes information related to the sections in the model cards proposal. We also largely excluded linked documents and papers from what we consider as the *primary documentation*. Our results should be interpreted with these decisions in mind. Whereas we could analyze a (near) complete set of models with model cards from corporations and GitHub, we had to sample for Hugging Face and baseline (GitHub models without model cards). Our samples were not truly random; to consider the impact of the models and

Table 2: Evaluation result for model documentation using our rubric. Each bar shows the percentage of model documentation that includes the information relating to the question. The vertical bar indicates the mean score across all data sources. The concrete rubric (question and their descriptions) is included in the supplementary material.

Question	Mean Score
Model Description	
Q1. Contact Information	
Q2. Model Type	
Q3. Model Date/Version	
Q4. Model License	
Intended Usages	
Q5. Intended Uses	
Q6. Out of Scope Uses	
Q7. How to Use	
Target Distribution	
Q8. Target Distribution Description	
Q9. Target Distribution Examples	
Evaluation Metrics	
Q10. Evaluation Statistics Reported	
Q11. Evaluation Statistics Explained	
Q12. Model Performance Visuals	
Evaluation Process	
Q13. Evaluation Process Explained	
Q14. Evaluation Data Explained	
Q15. Evaluation Data Available	
Training Process	
Q16. Training Process Explained	
Q17. Data Properties Explained	
Q18. Data Collection/Creation Explained	
Q19. Training Data Available	
Ethical Considerations	
Q20. Ethical Considerations Discussed	
Q21. Ethical Issue Mitigation Process	
Q22. Concrete Ethical Measurements	

to keep the analysis manageable, we stratified the sample among popular Hugging Face projects and focused on models for three tasks in the baseline. Finally, our analysis was manual and relies on some subjective judgment despite our best attempts to clarify and validate the rubric.

3.3 Assessment Results

3.3.1 Quantitative Result. In Table 2, we summarize what aspects are included in each model card and baseline model documentation in our dataset. Below, we discuss our observation on examining the documentation across sources and aspects in the model cards proposal. We use the term *model documentation* to refer to the entire documentation set we collected, including the baseline.

Model cards provided by companies and in GitHub repositories tend to include more information corresponding to model cards proposal than model cards on Hugging Face. Model cards provided by companies rank on top for 11 out of 22 questions based on the mean score, while Model cards in GitHub rank on top for eight questions (see Table 2). In contrast, model cards on Hugging Face are less likely to include information related to most questions - all but related to model type (Q2), model date and version (Q3), reporting evaluation statistics (Q10), and providing training data (Q19), by Dunn's Kruskal-Wallis multiple comparisons test. We also found no significant differences between the two strata of our Hugging Face sample, suggesting that the most popular models are not documented more comprehensively than less popular ones. Our baseline (documentation in GitHub repositories without mentioning model cards) draws a more mixed picture. They are rated similarly to or even higher than those of companies' model cards and GitHub model cards for some aspects, such as the intended use (Q5 and Q7), but fall short on questions related to ethical considerations (Q20-Q22), where they rarely include any information, similar to Hugging Face model cards.

Most model documentation have an unbalanced coverage of aspects in the model cards proposal. Overall, we found that 18 of the 22 types of information covered by our questions are included in less than half of the models' documentation. Only questions about the model type (Q2), model date and version (Q3), intended uses (Q5), and target distribution description (Q8) are included in more than half of the models' documentation. Q6 about the situations where a user should *not* use a model, however, is only documented in 32% of the models' documentation. Similarly, merely 35% of the models' documentation includes some discussion of bias or ethics (Q20). The ethical issue mitigation process (Q21) and measurement (Q22) are included in less than 10% of the models' documentation.

3.3.2 Further Observations. As discussed in Section 3.2.2, our rubric is used to evaluate the occurrence of information in the documentation for each question, not the comprehensiveness or correctness of the information. Nevertheless, during our assessment, we did observe strong variance in the extent to which the documentation answers those questions in the rubric.

The information provided in the model documentation is often vague. Taking Q8 about the *target distribution* as an example: Q8 is one of the few questions that is included in more than half of the model documentation. Yet, the majority of the documentation fails to provide more than very vague or generic information about the target distribution. For example, one model card[2] from Hugging Face claims to be a "Dutch pre-trained BERT model", hinting the target distribution being strings that represent text in Dutch – however, it leaves many questions about other characteristics of the inputs a model user could expect for the model to work for, such as the domain of the text (e.g., news, social media, reviews, etc.) and the style of the text (e.g., verbal and written). As one of the

[2]https://huggingface.co/GroNLP/bert-base-dutch-cased

few exceptions, Nvidia PeopleNet model card[3] represents a high-quality description of the target distribution, describing their model as being able to detect "persons, bags and faces" from "RGB Image of dimensions: 960 X 544 X 3 (W x H x C)." This model card then details the cases when the models might not perform as expected, including when target objects are smaller than 10x10 pixels, when more than 20% of the objects are occluded or truncated, when the photo is taken in dark lighting conditions, and so forth.

Shallow or no discussion along the ethical axes. While the GitHub projects model cards and the model cards from companies largely include information to answer Q20 (any ethical considerations), the information is often insufficient to examine concrete actual ethical issues related to the model. For example, one model card[4] from a company simply states in their ethical considerations section: *"we attempted to avoid bias and other ethical risks by not including demographic data in the model,"* but offer no further discussion, such as the scope of demographic data, the rationale of not including it, other biases from the data collection process, and steps taken for fairness auditing [34]. Such narrow or shallow discussions are reflected in the drop of scores between question Q20 (any ethical discussion) and questions Q21 and Q22 that engage with the ethical issue mitigation process and concrete measurements (see Table 2). A model card that demonstrates a more extensive explanation of ethical considerations can be seen in Salesforce's CTRL model card.[5] It mentions that the *"model was evaluated internally as well as externally by third parties, including the Partnership on AI, prior to release"* and provide a detailed description of steps they took to mitigate potential misuse, indicating that more extensive and meaningful effort towards mitigating potential ethical issues.

Model documentation often is not self-contained and sometimes directs the readers to additional resources. In particular, 45 out of the 132 model documentation (eight from Baseline and 37 from other model cards) in our dataset contain a link to an academic paper without summarizing the key information. In particular, 50% of the GitHub model documentation without mentioning model cards, and 44% of the Hugging Face model cards simply state that they are implementations or reimplementations of a specific linked academic paper. As discussed above, we do not consider this as an adequate substitute for the targeted and concise information suggested by the model cards proposal. A research paper is normally presented in a way targeting readers with sufficient academic background, not necessarily aligned with the background of the model users. Furthermore, in terms of comprehensiveness, including multiple links to outside sources decentralizes the information, increasing the chance of a model user missing key details.

3.4 Discussion

Drawing from our empirical investigations on model cards in the field, we discuss the gap between the original model cards proposal and the model documentation practice as well as its implications.

Top-down aspiration alone is insufficient to systematically improve the ML model documentation practice. Similar to the development teams of traditional software, the development teams of ML software constantly juggle different constraints and priorities [48]. While model documentation has been recognized as important by various stakeholders, the development team often sacrifices the effort of creating high-quality documentation for other seemingly more pressing concerns, such as a fast pace to the market. Our results also suggest that Hugging Face's attempt to promote the concept of model cards and to label the READMEs as model cards in their interface is ineffective to encourage the adoption of the model cards proposal. Indeed, on average, documentation of models on Hugging Face includes similar information to models published on GitHub without any mention of model cards. On the other hand, developers who voluntarily adopt model cards in GitHub repositories and those who publish models of companies tend to include information such as out-of-scope uses and evaluation results more systematically. Such effort might stem from the explicit requirements or intrinsic motivation for improving the model documentation.

Certain aspects of the ML models, in particular the aspects along the ethical axes, are rarely provided in the model cards in the field. The original model cards proposal aims to encourage model stakeholders to consider the broader context of model development and application. The ML documentation therefore should include the discussion about their decision making process during data collection and ethical consideration. However, our investigation reveals that even those who voluntarily adopt model cards commonly skip those sections. The ethical issue mitigation process (Q21) and measurement (Q22) were included in less than 10% of the models' documentation, suggesting only a shallow (public) engagement with fairness issues. These results demonstrate how most model documentation is insufficient for reasoning about the impact of model adoption, such as the model performance on unforeseen scenarios and on minority populations.

Meaningful encouragement for better documentation practices is needed during model development. As with any effective software engineering practice, the practice of documentation should be placed within the context of ML model development [56]; it will be ill-adopted otherwise. Hugging Face has provided the training materials for adopting the model cards proposal as part of their online tutorials. Nevertheless, our study reveals that the vast majority of its users do not take the time to follow the instructions provided; only a single one included ethical considerations or any discussion of bias. In fact, we found at least four Hugging Face model cards were created by the Hugging Face team on behalf of the model creators. Those findings indicate that ML model documentation in general still seems to be an afterthought at best. This is against the vision of the original model cards proposal that model cards should be used as an instrument to encourage more deliberate consideration of ethical aspects of ML models *during* model development. Based on those observations, we suggest that a more meaningful encouragement for better documenting ML models should start at the model development time rather than after.

4 DOCML DESIGN

Our work aims to accelerate wide adoption of model cards, improve compliance, and encourage more accountable documentation practice. In the meantime, we also acknowledge that ML model

[3]https://catalog.ngc.nvidia.com/orgs/nvidia/models/tlt_peoplenet
[4]https://help.salesforce.com/s/articleView?id=sf.mc_anb_einstein_messaging_insights_ethical_considerations.htm
[5]https://github.com/salesforce/ctrl/blob/master/ModelCard.pdf

development is a complex process that involves many different stakeholders, such as domain experts, data scientists, and software developers. As discussed in the previous section, an effective documentation tool should fit into the concrete workflow for individuals with different backgrounds. We, therefore, focus on the data scientists in this work as a starting point, considering their critical role in shaping model development. In this section, we present an interactive documentation tool that can be integrated as part of the model development routine of data scientists. We discuss the major considerations for designing such a tool and the implementation of our prototype.

4.1 DocML User Interface

The primary design goal of DocML is informed by our empirical analysis of model cards in the field that they are mainly ill-organized and missing important sections. DocML, therefore, aims to *nudge data scientists to consider and comply with the model cards proposal during the model development, especially the sections related to ethics that are often overlooked (design goal G1).*

We are also inspired by the existing body of literature on ML practices, in particular the discussion related to documentation. ML models are often improved in an iterative and continuous process [38, 48, 50, 59], mostly with additional data over time, there is a serious risk that model documentation and actual model properties drift apart. Since the documentation is critical for regulatory compliance, knowledge transfer, and reproducibility, its documentation quality needs to be constantly assessed and managed [28, 35]. While computational notebooks, the tools data scientists heavily rely on, have innate support for documentation as markdown cells, they are often messy and hard to make sense of [30]. DocML, therefore, aims to *support continuous assessing and managing the model documentation quality (design goal G2).*

Following the above two design goals, we built an early prototype of DocML. We then presented the prototype to users with a data science background from our connection and solicited additional feedback. Next, we further refined the prototype to integrate those additional design considerations. Below, we describe the final interface of DocML and discuss how it supports several scenarios when the data scientists develop and maintain their models and the corresponding documentation in the notebook environment.

4.1.1 Creating the model cards within the notebook environment (towards G1 and G2) . DocML is designed and implemented as an extension for JupyterLab, one of the most used notebook environments for data scientists. When activated, the interactive documentation panel will be expanded alongside their notebook on JupyterLab as shown in Figure 1. The pre-defined model card sections (introduced in Section 3.1) will show on this panel so that users will be reminded during the model development and documentation writing. Users can provide additional sections that reflecting their model development context, such as library use; they can also customize existing sections titles and orders, all through an explicit configuration file.

When users click the edit button next to the section title on the DocML panel, they can start filling in the content for that section. Instead of maintaining a separate data storage for the model card, we redirect users to an automatically created markdown cell in the notebook with the section title (shown as ① in Figure 2). To differentiate model card content that is more user oriented with other markdowns in the notebook that serve different purposes, we created a special set of HTML comments indicating their role in the model card. Users can view the latest version of model card under development anytime though clicking the *Refresh* button on the panel (see Figure 1). The completed model card can be exported to a markdown file for sharing by clicking the *Export to MD* button.

4.1.2 Nudging the adoption of model cards proposal (towards G1). To nudge data scientists to consider and to follow the model cards proposal more effectively, we designed and implemented several features. First, when the users hover their cursor on the title of each model card section, a concise description for that section is shown to remind them what content is appropriate (as indicated by ② in Figure 2). Moreover, explicit links to one or more examples can be added next to the section title so that users can reference when necessary (as indicated by ③ in the Figure 2). In this example, we select several high-quality model cards from our empirical study discussed in Section 3.3 as exemplars. The section titles, their descriptions, and examples links are also customizable through the configuration file.

Once the users finish editing the model card, they can export it to a markdown file for sharing. At this point, DocML will perform a lightweight completion check and inform them if any pre-defined sections are still empty (Figure 3). This extra step serves as an encouragement for them to complete missing sections.

4.1.3 Model card maintenance through code-document traceability (towards G2). Certain sections in the model card directly describe the purpose and outcome of the code cells in the notebook, such as the training process and evaluation process. The quality of those sections in this case depends on how accurate the code cells are described. To support the cross-reference between model card and source code, we adopt the concept of traceability, which is often used in software engineering for safety-critical systems [27]. In particular, users can explicitly link the code cells to the corresponding sections in the model cards so that the content can be easily referenced and analyzed during model card creation and maintenance. We defined six stages that represent a common machine learning pipeline related to the data scientists [3], i.e. data cleaning, preprocessing, hyperparameter tuning, model training, and model evaluation. To alleviate the manual effort required on selecting the stages, we automatically identify the stages for common libraries used by data scientists, including scikit-learn,[6] numpy,[7] pandas,[8] and matplotlib[9] through constructing a knowledge base of the API usage. Further support to other libraries can be added by enhancing the knowledge base. In case of incorrectly identified or missed code cells due to the auto-detection process, users can correct the stages manually. Under the hood, the trace links are maintained through the metadata for the code cells which cannot be easily viewed in the notebook environment. To make the information easily accessible, we automatically inject code comments to indicate those trace links, as shown in Figure 4.

[6] https://scikit-learn.org/stable/
[7] https://numpy.org
[8] https://pandas.pydata.org
[9] https://matplotlib.org

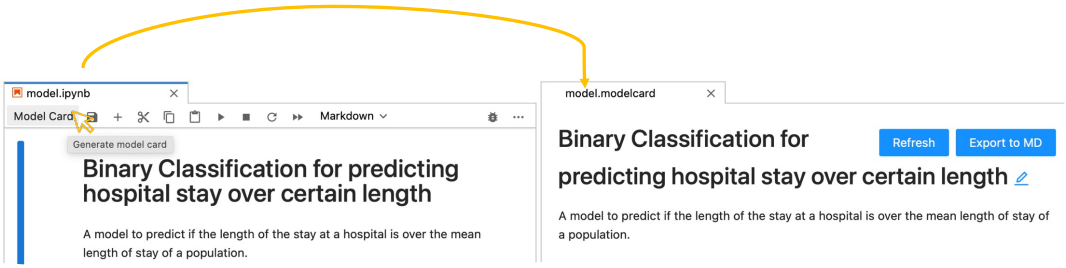

Figure 1: As the data scientists are developing models, they can activate DocML by clicking the tool button. The notebook and the documentation panel will show side by side on JupyterLab.

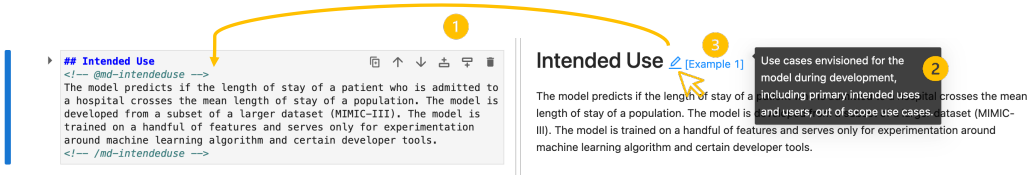

Figure 2: The user starts editing the model card for each section by clicking the edit button. The content is created and maintained within the notebook ①. DocML presents the description for each section when the cursor hovers over the title of the section ② and provides documentation examples through hyperlinks next to the section title ③.

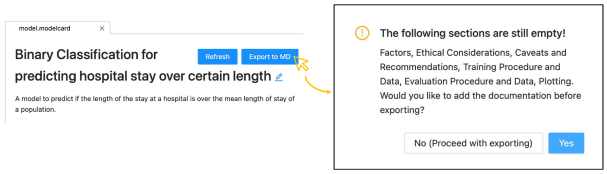

Figure 3: DocML suggests the users to add content in the empty sections before exporting.

Once the trace links are established, the model card maintainers or reviewers can easily navigate the code cells related to corresponding model card sections through a navigation bar on the DocML panel. It also indicates the relevant location of the code cell within the notebook (see Figure 5).

4.2 Implementation

The architecture of DocML includes a back-end extractor module, which is written in Python and JavaScript, and a front-end JupyterLab plugin written in React. It supports multiple model cards interfaces at the same time with the single back-end. The front-end module follows the interface design specifications described in Section 4.1. The back-end module analyzes the code in the notebook building on existing tools for program analysis,[10] ML stages analysis for notebook[11] and cell dependencies analysis.[12] Once a request is made from the front-end plugin, the content of the notebook is sent to the back-end for obtaining the cell dependencies and clustering the cells that belong to the same stage. Mappings to the relevant stage name are then added either based on the manual input by the user or based on knowledge base matching rules which classify various scikit-learn, numpy, pandas and matplotlib function calls in the sections.

The configuration file consists of a list of JSON objects, which record customizable content such as section names, their description, and any examples that showcase the suggested way of documentation. The JSON objects are populated and displayed in the front-end when DocML is initialized. All markdown cells from the Jupyter Notebook are parsed to retrieve any model documentation with the specific HTML tags for displaying on the tool panel.

5 USER STUDY

We conducted a lab user study to investigate to what degree DocML can support data scientists towards responsible and accountable documentation practice during model development and maintenance.

5.1 Study design

Ideally, we would have liked to observe how model developers use DocML in practice over an extended period to study the subtle effects of nudging and traceability. With the low adoption of model cards and few alternatives, we found such a design infeasible. Instead, we intentionally limited the scope of our study to questions we could ask within a controlled experiment involving experienced notebook users. In a lab setting, we cannot well study nudging effects on model cards, since we just freshly remind all participants (both in experimental and control group) about the model cards proposal. Hence we focus our study on exploring how

[10] https://github.com/andrewhead/python-program-analysis
[11] https://github.com/yjiang2cmu/Jupyter-Notebook-Project
[12] https://github.com/jerry-lu/cell-dependencies

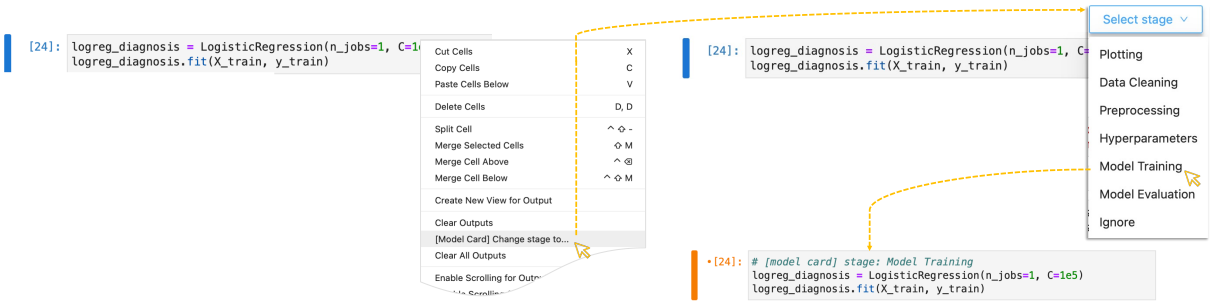

Figure 4: Users can explicitly select the machine learning stages corresponding to the model documentation. Once selected, the stage is indicated through a code comment.

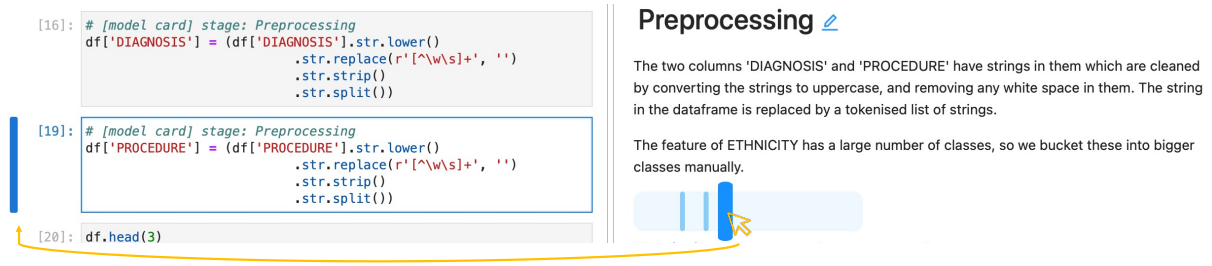

Figure 5: To support the users to navigate the corresponding cells in the notebook, DocML provide a navigation bar for each machine learning stage specified in the notebook.

DocML changes the focus and actions of its users, compared to users who are familiar with model cards but do not have dedicated tool support.

In a nutshell, we design a between-subject controlled experiment [6], where participants are given two notebooks with an incomplete, low-quality model card each and asked to perform tasks, including (a) selecting a model and (b) changing a model and its corresponding documentation. In the process, we observe how they navigate the notebooks, use the model cards, and make updates to the documentation. This allows us to focus on design aspects of our tool regarding tooling integration and transparency to answer the research questions below, but it provides only limited insight into the effectiveness of nudging in a natural environment – which we leave for future studies.

Our study aims to answer three research questions:

- **RQ1:** What kind of information does DocML encourage the data scientists to consider for ML model documentation?
- **RQ2:** What documentation approaches emerge when DocML is presented compared with the existing notebook environment?
- **RQ3:** What features could be changed or added to support data scientists in model documentation?

RQ1 relates to problems in current practices of using model cards (such as providing very selective documentation) and to our design goal *G1* of encouraging model developers to comply more with the model cards proposal, especially regarding ethics. In contrast, RQ2 relates primarily to our design goal *G2* of encouraging developers towards a process of continuous assessing and managing the model documentation. Finally, RQ3 seeks feedback on the tooling itself.

5.2 Method

We performed between-subject controlled experiments with 16 participants with or without DocML in a remote lab setting using Microsoft Teams. The study protocol was approved by the research ethics board at McGill University. Below, we discuss the details of the recruitment process, lab experiments design, and how we analyzed the experiments result.

5.2.1 Recruitment and Participants. We recruited participants through invitations on Twitter, LinkedIn, and from our personal networks followed by screening the candidates with appropriate backgrounds. The selection criteria included having sufficient experience with the notebook environment and having shared at least one ML model prior to the study. Suitable candidates were invited to participate in the study. In the end, 16 participants were recruited. The recruited participants were asked to fill out pre-study survey form, following which they participated in the remote lab study that lasts for around one hour, and then in a post-study interview.

Among them, 14 participants have at least one year of experience using computational notebooks. All participants have used Jupyter Notebook in academic settings (e.g. for assignments) and five of them also used Jupyter Notebook in professional settings (e.g. professional data scientists). The participants have a varied degree of experience with model documentation. While they have all developed and shared ML models with others before (the requirement for participating our study), four participants mentioned that

they have not written any explicit user-oriented documentation before. At the same time, one participant suggested that they have documented every model that they have developed.

5.2.2 Study Process. The 16 participants were randomly divided into two groups so that we can understand the impact introduced by our tool (RQ1 and RQ2). One group performed the study with DocML (the experimental condition; participants P_E1-P_E8), and the other group without our tool (control condition; participants P_C1-P_C8). In the pre-study survey, all participants answered questions about their data scientist background and documentation practice and were informed of the model cards proposal. Participants in the experimental group were additionally asked to watch a short tutorial video introducing the functions of DocML and to access and get familiar with DocML's interface.

At the start of the study, we provided the participants with an incomplete notebook and a model card in the form of a README.md file. This model card suffers from several common documentation quality issues. For example, it only contains a small number of model card sections, such as information about the model, intended use, and preprocessing. The information in each section is not necessarily complete. Moreover, the content of the model card is inconsistent with the original notebook in three places, specifically the library use, hyper-parameters, and the dataset feature description. We deliberately chose not to inform the participants of the concrete quality problems to mimic the model cards they might encounter in practice. All the study artifacts are included in the supplementary materials.

The participants from both groups were asked to perform two identical tasks, around 20 minutes each, representing common activities during ML model development and maintenance. Task 1 was to choose one among the two potential models we provided in the notebook and complete the documentation for the model of their choice. The participants were encouraged to make any changes to the existing code and documentation to improve their accuracy, completeness, or other quality attributes. During Task 2, the participants were asked to develop a new model on the same dataset using different features and to update the documentation accordingly. The resulting documentation from two groups was compared to answer RQ1. The entire process was video recorded for later analysis on their documentation activities to answer our RQ2.

Upon completion of the two tasks, we interviewed the participants about their experiences related to the model documentation (RQ3). For the experimental group, we asked the participants to evaluate six major features of DocML and how the features might fit in the workflow of data scientists. For the control group, we sought their opinion on the potential support that would improve their documentation experience.

5.2.3 Analysis. We analyze the experiment primarily qualitatively to understand how the participants approach documentation under different conditions. We first assessed the kind of changes they made to the provided model cards to answer RQ1. The analysis for RQ2 was done through a thematic analysis [66] of the video recordings by two of the authors. We particularly focused on the various activities they performed to understand data scientists' attitudes towards documentation creation and quality assessment and if they leverage the features of DocML when available.

5.3 Threat to Validity

The study is limited by the lab setting where the time constraint plays a major factor impacting the documentation experience of the participants. Task complexity, target domain, and other factors might play a bigger role in practice. Moreover, despite providing the tutorial and tool access prior to the study, participants in the experimental group still experience a learning curve of using the tool. Therefore, our study might not reflect the documentation quality developed by users who are already familiar with the interface. Finally, the long-term impact of deploying the tool, especially during continuous model and document evolution, cannot be observed from current study design.

5.4 Results and Observations

5.4.1 Consideration of the Model Cards Proposal (RQ1). **Observation of the control group.** Most participants from the control group made small edits on one or more sections in the provided model cards. The sections that were edited most include the *Hyperparameters* (P_C1, P_C2, P_C3, P_C6, P_C7) and *Preprocessing* (P_C3, P_C5, P_C6). Only P_C5 made changes on the section of *Intended Use*. Regarding the inconsistencies in the provided model card, P_C1 and P_C6 each fixed two places during the study, while the remaining six participants from the control group each fixed one.

In terms of the new content added, the participants mostly focused on different aspects of the model evaluation, including the performance metrics, evaluation process, and evaluation data. Some participants used more descriptive section titles such as "training procedure" (P_C8) and "test strategy" (P_C5) while other times participants used generic terms such as "model" (P_C1, P_C2, P_C3) and "result" (P_C4, P_C6).

Observation of the experimental group. The section edited most by the participants is *Data Cleaning* (P_E2, P_E3, P_E6, P_E8). Regarding the inconsistencies in the provided model card, P_E8 fixed two places during the study and the remaining seven participants from the experimental group each fixed one.

In terms of the new content, the sections to which the participants added the most are *Training Procedure and Data* (P_E2, P_E5, P_E7) and *Ethical Considerations* (P_E2, P_E4, P_E8). Notably, P_E2 added information about ethics including the sensitive features used by the model and the impact of the model if it is deployed to two specific sub-populations. Similarly, P_E4 and P_E8 added the use of race and ethnicity features in the model training. In comparison, no participants from the control group discussed ethics of the model development.

Despite providing the model cards template, participants sometimes still chose to add self-defined sections. For example, P_E1 added the *RandomForestClassifier* to describe the model type and features used. P_E3 and P_E5 added a section called *Exploratory Data Analysis* or *EDA* to document the data distribution. Both P_E4 and P_E6 added a *Problem Statement* in the markdown cells of the provided notebook describing the context of the model development in the model card. Occasionally, the newly added sections were named

using ambiguous terms such as *Results* (P_E3) and *Refined models* (P_E5).

When DocML was presented, the participants considered the scope of model documentation more broadly, as the model cards proposal suggested. Using the traditional notebook environment, the content of the model card was more performance-centric. In comparison, using DocML under time constraints, the participants tended to add fewer new sections to the model cards. However, the sections they added are more likely about the context of the model development, including the *ethical considerations* and *problem context* – the information often overlooked in public model documentation.

5.4.2 Documentation Approaches (RQ2). Participants from both groups generally started their tasks by glancing through the notebooks to get familiar with their structure and content. How they approached the documentation diverged depending on whether DocML was present. We identified three themes characterizing their documentation approaches when completing the model development and maintenance tasks. We describe them below and compare the differences when DocML was presented versus not.

Comparing and choosing different set of documentation. The notebook has innate support for documentation in the markdown cells. While the model card has a distinct purpose of presenting information about the ML model, its content inevitably is closely related to some of the markdown cells of the corresponding notebook. In Task 1, which was closer to a model card quality assessment setting, the participants needed to carefully compare and ensure the correctness of the information provided in the notebook and the model card. Participants from the control group, therefore, spent considerable time comparing the notebook markdown cells and provided model cards. If they spotted any problems, they had to choose *where* to fix them. While occasionally they made changes in one set of documentation and copied the content over to the other, most of the time they simply changed the model card but left the markdown cells unchanged, leading to inconsistent information (all except P_C3 and P_C6). In contrast, the problem of inconsistency between the two sets of documentation was naturally eliminated in the experimental group when they used DocML, since DocML ensured that the documentation for model cards would always be added to the notebook and updated in the model card view (all except P_E5 and P_E8).[13] In Task 2, which mimicked a model card development setting, we observe similar behaviors for the participants in the control group. They either added the documentation to the notebook and but not the model card (P_C1, P_C4, P_C8), or first added the content into the markdown file and then copied it to the model card (P_C2, P_C3, P_C4, P_C5). P_C4 even copied some code snippets into the model card. On the other hand, most participants in the experimental group (all except P_E7 and P_E8) ensured that the same documentation that was added to the notebook was also present in the model card by using DocML.

Locating corresponding source code. Some of the model card sections directly describe the source code in the notebook and its outcome. The code can be scattered into multiple cells that are disconnected. We observe that the participants in the control group spent a significant amount of effort on locating the code cells when inspecting the provided model cards during the maintenance task. P_C7, for example, had a hard time finding the code cells corresponding to the sections related to dataset cleaning, pre-processing, and hyperparameters, and had to scroll the entire notebook several times before starting to settle on some of the code cells. P_C1, P_C2, P_C3, P_C4 and P_C5 showed a similar struggle. On the other hand, all participants except P_E8 from the experimental group used the DocML's navigation support enabled by the code-documentation trace links to help them examine the model card content. The participants either heavily relied on the navigation bar from the start (P_E7, P_E9) or increased their usage of the bar for finding the code cells as they got more used to the tool (P_E1, P_E3, P_E4, P_E5).

Devoting attention during documentation. In the control group, participants paid most of their attention to examining the existing notebook and the provided model card during Task 1. Almost all participants in this group (except P_C8) devoted their effort to adding the documentation related to the algorithmic aspect of the model and its training or testing processes. When DocML was presented, all participants from the experimental group except P_E1 and P_E3 read the prompt descriptions of model card sections to help their understanding, in particular, the sections of *Factors*, *Fairness considerations*, and *Caveat and Recommendation*. Some of them further clicked the example links of model cards. The difference in attention between the two groups explains the observed differences in the resulting model cards in RQ1 – the model cards from the control group heavily emphasized the model evaluation whereas the model cards from the experimental group focused more on the model development context and ethical considerations.

Additionally, DocML seems to noticeably influence the participants to consider the trace links as an innate component of the documentation both during model maintenance and development. Some of them (P_E5, P_E6, P_E7) made edits to the trace links by modifying the existing links from code cells and/or adding new links for Task 1. For Task 2 during which they were asked to develop their own models, P_E6 and P_E7 devoted considerable effort to creating trace links for their newly added code cells. This additional effort might be motivated by the experienced benefit of code-documentation trace links in Task 1.

DocML considerably alleviates the effort of assessing and managing the model documentation quality and prompts the participants towards more accountable documentation practice. In contrast to the control group, most participants from the experimental group chose to put more documentation effort into activities that are missing from the current practice, including understanding the model cards proposal (especially ethical-related considerations) and creating and maintaining doc-code trace links. Those activities can potentially bring non-negligible benefits to model documentation in the long term.

5.4.3 User Evaluation and Feedback (RQ3). **Feedback from the experimental group.** Figure 6 summarizes how the participants from the experimental group rated the necessity and ease of use for different features in DocML.

[13] Due to a technical issue during the study, DocML only became accessible to P_E8 at the end of Task 2.

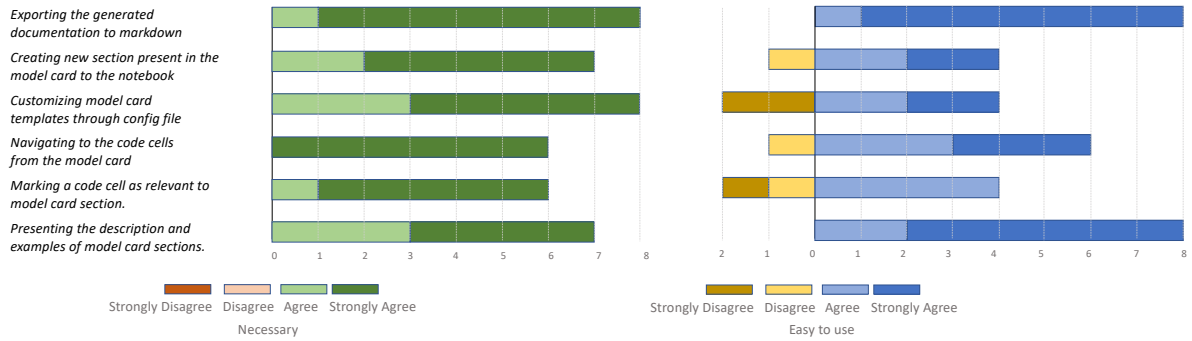

Figure 6: User evaluation on the necessity and ease of use for DocML. Both aspects were evaluated through a Likert scale. Neutral and abstained input is omitted from the figure. X axis is the number of participants from the experimental group.

Most participants agreed or strongly agreed that the features provided by DocML are necessary. All participants except P_E7, in particular, commented on the importance of having the model card template to structure their documentation **during** the model development. They thought the prompted descriptions were helpful to better understand the model card sections. Moreover, as suggested by P_E1, P_E2, and P_E3, DocML offers a preview of the model card during development time that is effectively separated from the notebook markdown cells. It enables the participants to consider how the information is communicated to the users. At the same time, P_E5 appreciated how DocML can prevent inconsistency between the developmental documentation and model cards.

Most participants (all participants expect P_E5 and P_E8) also thought the navigation function supported by the trace links was especially effective so that they *"do not have to see hundreds of lines of code for going to the [model card] section"* (P_E2). On the other hand, since the current construction of trace links requires several mouse clicks to find the stages in the model development pipeline (see Figure 4), P_E1 and P_E3 suggested having more intuitive options or automated solutions to achieve similar functions.

At the same time, participants from the experimental group expected more control over the section order and title in the model card. They preferred direct modification on the model card template through the UI panel rather than the configuration file, indicating a tension between customization and standardization that needs to be carefully balanced in practice. Moreover, the participants hoped the markdown cells representing the model card content could be injected next to the corresponding code cell. DocML currently is limited by depending on the users to appropriately locate the newly added markdown in the notebook.

Feedback from the control group. When asked what features to expect for a model documentation tool in the notebook environment, participants from the control group voiced their needs for many similar functions provided by DocML, such as the documentation template (P_C2, P_C3, P_C8), documentation extraction from the markdown cells or code cells (P_C1, P_C4, P_C7), and code-documentation links (P_C3, P_C6). Those suggestions stem from both their experience during our user study and their previous experience as data scientists (and software developers). P_C3 and P_C4 specifically compared the documentation support for data scientists with more traditional software developers and pointed out that more mature tools and frameworks such as JavaDoc [49] and Sphinx [18] are unavailable. Such lack of support has caused them the most frustration during documenting models.

Participants from the experimental group judged the functions of DocML were both necessary and generally easy to use. At the same time, they expected improvement in flexibility and more intuitive switches between the notebook and the model card panels.

6 DISCUSSION

In this work, we systematically investigated how the model cards proposal has been adopted in the field, finding a substantial gap between the ambitions behind model cards and actual practice, finding that model cards are rare and often shallow. Motivated by this gap and guided by literature on ML practices, we designed DocML to provide data scientists direct documentation support to follow the model cards proposal and other best practices during model development and maintenance. Here, we discuss the most important findings of our work, their broad implications and limitations.

6.1 Aspirations vs. Practice

In all of GitHub with millions of public notebooks [51, 53, 57] and many repositories sharing learning code and learned models, we found only 24 models documented explicitly with model cards. Our best effort in finding model cards published by companies also resulted in only 28 models. Considering that the model card paper is one of the most cited works on ML documentation and is often recommended, such a limited adoption indicates reluctance or difficulty to transform recommendations into standard practice.

Furthermore, even when model cards were adopted as a concept and term, they were often of low quality and provided only selective information. During our assessment, we observed strong variance in what information was provided by the model cards. Moreover, the extent to which the documentation answers those questions in the rubric also varies drastically. For example, the majority of the model cards we examined failed to provide more than vague or generic information related to target distribution and ethical considerations. Model cards found in practice were often not

self-contained and sometimes directed the readers to additional resources such as research papers, thus not fulfilling the intention of providing a concise place for essential documentation. Even when digging into the papers, we often failed to find information related to the various recommended model card sections. In general, model documentation in practice seems to still be an afterthought at best.

The original model cards paper lists several important roles that model cards serve, including improving model understanding, helping *"to standardize decision making processes for invested stakeholders,"* and encouraging *"forward-looking model analysis techniques"* [45]. We argue that **the current state of model documentation with model cards fulfills none of these roles well.** Previous work on interactive model cards [16] provides an example of adopting model cards to improve the model understanding, whereas our tool DocML attempts to encourage better adoption to support the other two roles of the model cards. Further studies are needed in understanding and assisting the multifaceted purpose of model cards in practice and ML documentation in general.

6.2 Design to Facilitate Documentation Activities

The design of documentation tools should build on the consideration of the unique characteristics of ML documentation, including how it relates to source code development and its important quality attributes. The existing *Model Card Toolkit* [26] was proposed in 2020 but has barely received any adoption on GitHub. We conjecture that this is due to a mismatch between tool focus and model card needs: The Model Card Toolkit is useful to report statistics and evaluation metrics directly from the source code (similar to experiment tracking tools like MLflow[14] or Neptune[15]), but it provides little value for the many other important sections in model cards that require users to manually provide information about intentions, concerns, and ethical deliberations.

We build on the insight that **nudging and traceability in an integrated development environment afford better documentation practices.** In our user study, we observed that when our tool was not available, data scientists were overwhelmed with navigating and documenting details of the model development code, leaving no room for considering the model cards, despite the awareness of such a recommendation. In comparison, when the documentation environment was integrated with the coding environment in a meaningful way, data scientists were devoting more effort to improving documentation quality and maintainability. They approached documentation in a more iterative manner and actively used and maintained the trace links between documentation and the source code. The data scientists further spend more time considering the context and impact of the model development and deployment when the explanation and examples of model card sections are nudged in their model development environment. We argue that those activities are critical to the comprehensiveness of model documentation, in particular along the ethical axis.

Among the major features of DocML, nudging is a familiar concept for the CHI community and has been already equipped in the digital interface designers' toolbox. Recent work by Caraban et al. categorizes nudging mechanisms in technology design for health, sustainability, and privacy [11]. The six categories are facilitate, confront, deceive, social influence, fear, and reinforce. Among them, facilitate (e.g., default options) and reinforce (e.g., just-in-time prompts) have been adopted for encouraging software engineering behaviors [9, 39]. Similarly, our work uses *facilitate* and *reinforce* mechanisms in the documentation tool for the model developers to more consciously consider and document the model usage and ethical issues during development. We invite researchers to investigate the potential of other mechanisms (such as *confront* by reminding the consequences of not completing the documentation sections) to extend the nudging effect into a broader context.

The concept of traceability and how to approach it through design, however, are less discussed in the human-computer interaction literature. In the context of software development, traceability is an important property for developing safety-critical software [27]. It is often required by regulatory bodies to demonstrate the quality of the software development process and the resulting software. Traceability is also suggested to improve AI accountability by Raji et al. [54]. In our tool DocML, explicit traceability links between code and documentation can support examining the consistency between those two sets of artifacts and therefore improve the accuracy of the documentation and the accountability of the machine learning models. Despite not using the same terminology of traceability, recent work on data documentation also proposes to treat "text-data connections as persistent, interactive, first class objects" [13]. We hope our work and similar attempts can draw more attention from the community to understanding and making use of traceability links during documentation activities that can greatly contribute to the quality and accountability of ML documentation and their systems as a whole.

6.3 Limitation and Future Work

As discussed in Section 5.1, our user study is limited by the lab setting with concrete instructions on the model development tasks that are small in scope. Moreover, we did not explicitly ask the participants about their thoughts on the model cards proposal, but simply informed them about this proposal through the pre-study survey. In future work, we plan to study the impact of DocML in practice with more complex development tasks and other practical concerns (such as model domain and team culture). It is also important to consider how the documentation support should evolve as data scientists gain experience with the model cards proposal. Additionally, while our current work focuses on supporting documentation tasks for data scientists in the notebook environment, future work can be expanded on streamlining documentation activities for their entire workflow that rely on various tools across different environments.

In our study, we observed how data scientists face difficulty in interpreting the implications of ML models outside the scope of the model development pipeline. This challenge can be amplified in practice. When building ML-enabled software products, an ML model contributes to the overall product but is just one component among many in a system. Such products are typically built by interdisciplinary teams, where software engineers and UI experts integrate ML models into a larger software project, considering

[14] https://mlflow.org/
[15] https://neptune.ai/home

many non-ML concerns. Previous work suggested that, in many projects, data scientists building the model are siloed off and rarely interact with teams using their models, causing many conflicts and misunderstandings at the interface between the teams [48]. Modern MLOps practices and the corresponding rise of the role of ML engineers [40, 58], who focus primarily on automating machine learning pipelines, supporting experimentation, and deploying and updating models, introduce further complexities and roles. Model cards, in this case, should not be authored by the data scientists alone. Instead, they should serve as a shared important artifact between the teams that captures the wide range of concerns. In future work, we will examine the potential of considering model cards as *boundary objects* that are used to negotiate and communicate between data scientists and software engineers to support collaborative interdisciplinary work [41, 61].

7 CONCLUSION

In this work, we investigated how publicly available ML models are documented, especially when they adopted the model cards proposal. Our assessment of those model cards reveals a clear gap between the proposal and practice. In an effort to move the needle towards meaningful adoption of the model cards proposal and improving documentation practice, we proposed a model documentation tool DocML for data scientists using computational notebooks. As demonstrated in the user study, when the DocML was presented, data scientists more actively improved documentation and considered ethical implications during model development. They also spent considerable effort on the construction and maintenance of trace links between documentation and source code, which supports model accountability. Our work highlights the new opportunities of designing machine learning documentation for long-term benefit through nudging and traceability.

ACKNOWLEDGMENTS

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) RGPIN- 2021-03538, the National Science Foundation (NSF) award 1813598 and 2131477, and Fonds de recherche du Québec (FRQNT) 2021-NC-284820. Coursey participated in this work as part of the NSF-supported REU-SE program. We thank our anonymous participants. We also thank our lab members and the CHI reviewers for their constructive feedback.

REFERENCES

[1] Emad Aghajani, Csaba Nagy, Mario Linares-Vásquez, Laura Moreno, Gabriele Bavota, Michele Lanza, and David C. Shepherd. 2020. Software Documentation: The Practitioners' Perspective. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering* (Seoul, South Korea) *(ICSE '20)*. Association for Computing Machinery, New York, NY, USA, 590–601. https://doi.org/10.1145/3377811.3380405

[2] Emad Aghajani, Csaba Nagy, Olga Lucero Vega-Márquez, Mario Linares-Vásquez, Laura Moreno, Gabriele Bavota, and Michele Lanza. 2019. Software Documentation Issues Unveiled. In *Proceedings of the 41st International Conference on Software Engineering* (Montreal, Quebec, Canada) *(ICSE '19)*. IEEE Press, 1199–1210. https://doi.org/10.1109/ICSE.2019.00122

[3] Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. 2019. Software Engineering for Machine Learning: A Case Study. In *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice* (Montreal, Quebec, Canada) *(ICSE-SEIP '19)*. IEEE Press, 291–300. https://doi.org/10.1109/ICSE-SEIP.2019.00042

[4] Matthew Arnold, Rachel KE Bellamy, Michael Hind, Stephanie Houde, Sameep Mehta, Aleksandra Mojsilović, Ravi Nair, K Natesan Ramamurthy, Alexandra Olteanu, David Piorkowski, et al. 2019. FactSheets: Increasing trust in AI services through supplier's declarations of conformity. *IBM Journal of Research and Development* 63, 4/5 (2019), 6–1.

[5] Deeksha M Arya, Jin LC Guo, and Martin P Robillard. 2020. Information correspondence between types of documentation for APIs. *Empirical Software Engineering* 25, 5 (2020), 4069–4096.

[6] Ann Blandford, Anna L. Cox, and Paul Cairns. 2008. *Controlled experiments*. Cambridge University Press, 1–16. https://doi.org/10.1017/CBO9780511814570.002

[7] Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (Technology) is Power: A Critical Survey of "Bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 5454–5476. https://doi.org/10.18653/v1/2020.acl-main.485

[8] Karen L Boyd. 2021. Datasheets for Datasets help ML Engineers Notice and Understand Ethical Issues in Training Data. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–27.

[9] Chris Brown and Chris Parnin. 2021. Nudging Students Toward Better Software Engineering Behaviors. In *2021 IEEE/ACM Third International Workshop on Bots in Software Engineering (BotSE)*. IEEE Press, 11–15. https://doi.org/10.1109/BotSE52550.2021.00010

[10] Joy Buolamwini and Timnit Gebru. 2018. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency (Proceedings of Machine Learning Research, Vol. 81)*, Sorelle A. Friedler and Christo Wilson (Eds.). PMLR, 77–91. https://proceedings.mlr.press/v81/buolamwini18a.html

[11] Ana Caraban, Evangelos Karapanos, Daniel Gonçalves, and Pedro Campos. 2019. *23 Ways to Nudge: A Review of Technology-Mediated Nudging in Human-Computer Interaction*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3290605.3300733

[12] Souti Chattopadhyay, Ishita Prasad, Austin Z. Henley, Anita Sarma, and Titus Barik. 2020. What's Wrong with Computational Notebooks? Pain Points, Needs, and Design Opportunities. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376729

[13] Zhutian Chen and Haijun Xia. 2022. CrossData: Leveraging Text-Data Connections for Authoring Data Documents. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 95, 15 pages. https://doi.org/10.1145/3491102.3517485

[14] Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20, 1 (1960), 37–46. https://doi.org/10.1177/001316446002000104 arXiv:https://doi.org/10.1177/001316446002000104

[15] Luis Fernando Cortés-Coy, Mario Linares-Vásquez, Jairo Aponte, and Denys Poshyvanyk. 2014. On Automatically Generating Commit Messages via Summarization of Source Code Changes. In *2014 IEEE 14th International Working Conference on Source Code Analysis and Manipulation*. IEEE Press, 275–284. https://doi.org/10.1109/SCAM.2014.14

[16] Anamaria Crisan, Margaret Drouhard, Jesse Vig, and Nazneen Rajani. 2022. Interactive Model Cards: A Human-Centered Approach to Model Documentation. In *2022 ACM Conference on Fairness, Accountability, and Transparency*. ACM, Seoul Republic of Korea, 427–439. https://doi.org/10.1145/3531146.3533108

[17] Jeffrey Dastin. 2018. *Amazon scraps secret AI recruiting tool that showed bias against women*. https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G

[18] The Sphinx developers. [n.d.]. *Welcome: Sphinx makes it easy to create intelligent and beautiful documentation.*

[19] Julia Dressel and Hany Farid. 2018. The accuracy, fairness, and limits of predicting recidivism. *Science advances* 4, 1 (2018), eaao5580.

[20] Ian Drosos, Titus Barik, Philip J. Guo, Robert DeLine, and Sumit Gulwani. 2020. *Wrex: A Unified Programming-by-Example Interaction for Synthesizing Readable Code for Data Scientists*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376442

[21] Hugging Face. 2021. Hugging Face Documentation: Building a model card. Retrieved Jan 5, 2022 from https://huggingface.co/course/chapter4/4

[22] Hugging Face. 2021. Hugging Face Documentation: Model Repos docs. Retrieved Jan 5, 2022 from https://huggingface.co/docs/hub/model-repos

[23] Hugging Face. 2021. Hugging Face – The AI community building the future. Retrieved July 20, 2021 from https://huggingface.co/

[24] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for Datasets. *Commun. ACM* 64, 12 (nov 2021), 86–92. https://doi.org/10.1145/3458723

[25] Joan Giner-Miguelez, Abel Gómez, and Jordi Cabot. 2022. DescribeML: A Tool for Describing Machine Learning Datasets. In *Proceedings of the 25th International*

Conference on Model Driven Engineering Languages and Systems: Companion Proceedings (Montreal, Quebec, Canada) *(MODELS '22)*. Association for Computing Machinery, New York, NY, USA, 22–26. https://doi.org/10.1145/3550356.3559087

[26] Google. 2020. model-card-toolkit. https://github.com/tensorflow/model-card-toolkit.

[27] Orlena Gotel, Jane Cleland-Huang, Jane Huffman Hayes, Andrea Zisman, Alexander Egyed, Paul Grünbacher, Alex Dekhtyar, Giuliano Antoniol, Jonathan Maletic, and Patrick Mäder. 2012. *Traceability Fundamentals*. Springer London, London, 3–22. https://doi.org/10.1007/978-1-4471-2239-5_1

[28] Mark Haakman, Luís Cruz, Hennie Huijgens, and Arie van Deursen. 2021. AI Lifecycle Models Need to Be Revised: An Exploratory Study in Fintech. *Empirical Softw. Engg.* 26, 5 (sep 2021), 29. https://doi.org/10.1007/s10664-021-09993-1

[29] Andrew Head, Fred Hohman, Titus Barik, Steven M. Drucker, and Robert DeLine. 2019. Managing Messes in Computational Notebooks. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300500

[30] Andrew Head, Fred Hohman, Titus Barik, Steven M. Drucker, and Robert DeLine. 2019. Managing Messes in Computational Notebooks. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300500

[31] Andrew Head, Jason Jiang, James Smith, Marti A. Hearst, and Björn Hartmann. 2020. Composing Flexibly-Organized Step-by-Step Tutorials from Linked Source Code, Snippets, and Outputs. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376798

[32] Jazlyn Hellman, Eunbee Jang, Christoph Treude, Chenzhun Huang, and Jin LC Guo. 2021. Generating GitHub Repository Descriptions: A Comparison of Manual and Automated Approaches. (2021). https://doi.org/10.48550/arXiv.2110.13283 arXiv:arXiv:2110.13283

[33] Sarah Holland, Ahmed Hosny, and Sarah Newman. 2020. The dataset nutrition label. *Data Protection and Privacy, Volume 12: Data Protection and Democracy* 12 (2020), 1.

[34] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé, Miro Dudik, and Hanna Wallach. 2019. *Improving Fairness in Machine Learning Systems: What Do Industry Practitioners Need?* Association for Computing Machinery, New York, NY, USA, 1–16. https://doi.org/10.1145/3290605.3300830

[35] Aspen Hopkins and Serena Booth. 2021. Machine Learning Practices Outside Big Tech: How Resource Constraints Challenge Responsible Development. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society* (Virtual Event, USA) *(AIES '21)*. Association for Computing Machinery, New York, NY, USA, 134–145. https://doi.org/10.1145/3461702.3462527

[36] Geoff Hulten. 2018. *Building Intelligent Systems: A Guide to Machine Learning Engineering*. Apress.

[37] Jigsaw. 2017. Perspective API. https://www.perspectiveapi.com/.

[38] Mary Beth Kery, Marissa Radensky, Mahima Arya, Bonnie E. John, and Brad A. Myers. 2018. The Story in the Notebook: Exploratory Data Science Using a Literate Programming Tool. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3173574.3173748

[39] Douglas Kramer. 1999. API Documentation from Source Code Comments: A Case Study of Javadoc. In *Proceedings of the 17th Annual International Conference on Computer Documentation* (New Orleans, Louisiana, USA) *(SIGDOC '99)*. Association for Computing Machinery, New York, NY, USA, 147–153. https://doi.org/10.1145/318372.318577

[40] Valliappa Lakshmanan, Sara Robinson, and Michael Munn. 2020. *Machine learning design patterns*. O'Reilly Media.

[41] Charlotte P. Lee. 2005. Between Chaos and Routine: Boundary Negotiating Artifacts in Collaboration. In *ECSCW 2005*, Hans Gellersen, Kjeld Schmidt, Michel Beaudouin-Lafon, and Wendy Mackay (Eds.). Springer Netherlands, Dordrecht, 387–406.

[42] Z. Liu, P. Luo, X. Wang, and X. Tang. 2015. Deep Learning Face Attributes in the Wild. In *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA, 3730–3738. https://doi.org/10.1109/ICCV.2015.425

[43] Katherine R. Maffey, Kyle Dotterrer, Jennifer Niemann, Iain Cruickshank, Grace A. Lewis, and Christian Kästner. 2023. MLTEing Models: Negotiating, Evaluating, and Documenting Model and System Qualities. In *Proc. International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*.

[44] Paul W. McBurney and Collin McMillan. 2014. Automatic Documentation Generation via Source Code Summarization of Method Context. In *Proceedings of the 22nd International Conference on Program Comprehension* (Hyderabad, India) *(ICPC 2014)*. Association for Computing Machinery, New York, NY, USA, 279–290. https://doi.org/10.1145/2597008.2597149

[45] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model Cards for Model Reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (Atlanta, GA, USA) *(FAT* '19)*. Association for Computing Machinery, New York, NY, USA, 220–229. https://doi.org/10.1145/3287560.3287596

[46] Laura Moreno, Jairo Aponte, Giriprasad Sridhara, Andrian Marcus, Lori Pollock, and K. Vijay-Shanker. 2013. Automatic generation of natural language summaries for Java classes. In *2013 21st International Conference on Program Comprehension (ICPC)*. IEEE Press, 23–32. https://doi.org/10.1109/ICPC.2013.6613830

[47] Michael Muller, Ingrid Lange, Dakuo Wang, David Piorkowski, Jason Tsay, Q. Vera Liao, Casey Dugan, and Thomas Erickson. 2019. How Data Science Workers Work with Data: Discovery, Capture, Curation, Design, Creation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3290605.3300356

[48] Nadia Nahar, Shurui Zhou, Grace Lewis, and Christian Kästner. 2022. Collaboration Challenges in Building ML-Enabled Systems: Communication, Documentation, Engineering, and Process. In *Proceedings of the 44th International Conference on Software Engineering* (Pittsburgh, Pennsylvania) *(ICSE '22)*. Association for Computing Machinery, New York, NY, USA, 413–425. https://doi.org/10.1145/3510003.3510209

[49] Oracle. [n.d.]. How to Write Doc Comments for the Javadoc Tool. https://www.oracle.com/ca-en/technical-resources/articles/java/javadoc-tool.html.

[50] Kayur Patel, James Fogarty, James A. Landay, and Beverly Harrison. 2008. Investigating Statistical Machine Learning as a Tool for Software Development. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 667–676. https://doi.org/10.1145/1357054.1357160

[51] João Felipe Pimentel, Leonardo Murta, Vanessa Braganholo, and Juliana Freire. 2019. A Large-Scale Study about Quality and Reproducibility of Jupyter Notebooks. In *Proceedings of the 16th International Conference on Mining Software Repositories* (Montreal, Quebec, Canada) *(MSR '19)*. IEEE Press, 507–517. https://doi.org/10.1109/MSR.2019.00077

[52] Gede Artha Azriadi Prana, Christoph Treude, Ferdian Thung, Thushari Atapattu, and David Lo. 2019. Categorizing the content of github readme files. *Empirical Software Engineering* 24, 3 (2019), 1296–1327.

[53] Fotis Psallidas, Yiwen Zhu, Bojan Karlas, Matteo Interlandi, Avrilia Floratou, Konstantinos Karanasos, Wentao Wu, Ce Zhang, Subru Krishnan, Carlo Curino, and Markus Weimer. 2019. Data Science through the looking glass and what we found there. (2019). https://doi.org/10.48550/ARXIV.1912.09536 arXiv:arXiv:1912.09536

[54] Inioluwa Deborah Raji, Andrew Smart, Rebecca N. White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. 2020. Closing the AI Accountability Gap: Defining an End-to-End Framework for Internal Algorithmic Auditing. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (Barcelona, Spain) *(FAT* '20)*. Association for Computing Machinery, New York, NY, USA, 33–44. https://doi.org/10.1145/3351095.3372873

[55] John Richards, David Piorkowski, Michael Hind, Stephanie Houde, and Aleksandra Mojsilović. 2020. A Methodology for Creating AI FactSheets. https://doi.org/10.48550/ARXIV.2006.13796 arXiv:arXiv:2006.13796

[56] Martin P. Robillard, Andrian Marcus, Christoph Treude, Gabriele Bavota, Oscar Chaparro, Neil Ernst, Marco Aurélio Gerosa, Michael Godfrey, Michele Lanza, Mario Linares-Vásquez, Gail C. Murphy, Laura Moreno, David Shepherd, and Edmund Wong. 2017. On-demand Developer Documentation. In *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE Press, 479–483. https://doi.org/10.1109/ICSME.2017.17

[57] Adam Rule, Aurélien Tabard, and James D. Hollan. 2018. Exploration and Explanation in Computational Notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3173606

[58] D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. 2015. Hidden Technical Debt in Machine Learning Systems. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2015/file/86df7dcfd896fcaf2674f757a2463eba-Paper.pdf

[59] Julien Siebert, Lisa Joeckel, Jens Heidrich, Adam Trendowicz, Koji Nakamichi, Kyoko Ohashi, Isao Namba, Rieko Yamamoto, and Mikio Aoyama. 2022. Construction of a quality model for machine learning systems. *Software Quality Journal* 30, 2 (2022), 307–335.

[60] Ian Sommerville. 2016. *Software Engineering, 10th edition*. Pearson.

[61] Susan Leigh Star and James R Griesemer. 1989. Institutional ecology,translations' and boundary objects: Amateurs and professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39. *Social studies of science* 19, 3 (1989), 387–420.

[62] Kathryn T. Stolee, Sebastian Elbaum, and Matthew B. Dwyer. 2016. Code search with input/output queries: Generalizing, ranking, and assessment. *Journal of Systems and Software* 116 (2016), 35–48. https://doi.org/10.1016/j.jss.2015.04.081

[63] J. Stylos and B.A. Myers. 2006. Mica: A Web-Search Tool for Finding API Components and Examples. In *Visual Languages and Human-Centric Computing (VL/HCC'06)*. IEEE Press, 195–202. https://doi.org/10.1109/VLHCC.2006.32

[64] Christoph Treude and Martin P. Robillard. 2016. Augmenting API Documentation with Insights from Stack Overflow. In *Proceedings of the 38th International Conference on Software Engineering* (Austin, Texas) *(ICSE '16)*. Association for Computing Machinery, New York, NY, USA, 392–403. https://doi.org/10.1145/2884781.2884800

[65] Gias Uddin and Martin P. Robillard. 2015. How API Documentation Fails. *IEEE Software* 32, 4 (2015), 68–75. https://doi.org/10.1109/MS.2014.80

[66] Mojtaba Vaismoradi, Hannele Turunen, and Terese Bondas. 2013. Content analysis and thematic analysis: Implications for conducting a qualitative descriptive study. *Nursing & Health Sciences* 15, 3 (2013), 398–405. https://doi.org/10.1111/nhs.12048 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/nhs.12048

[67] April Yi Wang, Will Epperson, Robert A DeLine, and Steven M. Drucker. 2022. Diff in the Loop: Supporting Data Comparison in Exploratory Data Analysis. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 97, 10 pages. https://doi.org/10.1145/3491102.3502123

[68] April Yi Wang, Dakuo Wang, Jaimie Drozdal, Michael Muller, Soya Park, Justin D. Weisz, Xuye Liu, Lingfei Wu, and Casey Dugan. 2022. Documentation Matters: Human-Centered AI System to Assist Data Science Code Documentation in Computational Notebooks. *ACM Trans. Comput.-Hum. Interact.* 29, 2, Article 17 (jan 2022), 33 pages. https://doi.org/10.1145/3489465

[69] Nathaniel Weinman, Steven M. Drucker, Titus Barik, and Robert DeLine. 2021. Fork It: Supporting Stateful Alternatives in Computational Notebooks. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 307, 12 pages. https://doi.org/10.1145/3411764.3445527

[70] Chenyang Yang, Shurui Zhou, Jin L. C. Guo, and Christian Kästner. 2022. Subtle Bugs Everywhere: Generating Documentation for Data Wrangling Code. In *Proceedings of the 36th IEEE/ACM International Conference on Automated Software Engineering* (Melbourne, Australia) *(ASE '21)*. IEEE Press, 304–316. https://doi.org/10.1109/ASE51524.2021.9678520

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