Model Sketching: Centering Concepts in Early-Stage Machine Learning Model Design

Michelle S. Lam
Stanford University
Stanford, CA, USA
mlam4@cs.stanford.edu

Zixian Ma
Stanford University
Stanford, CA, USA
zixianma@cs.stanford.edu

Anne Li
Stanford University
Stanford, CA, USA
anne24@stanford.edu

Izequiel Freitas
Stanford University
Stanford, CA, USA
ifreitas@stanford.edu

Dakuo Wang*
Northeastern University
Boston, MA, USA
d.wang@northeastern.edu

James A. Landay
Stanford University
Stanford, CA, USA
landay@stanford.edu

Michael S. Bernstein
Stanford University
Stanford, CA, USA
msb@cs.stanford.edu

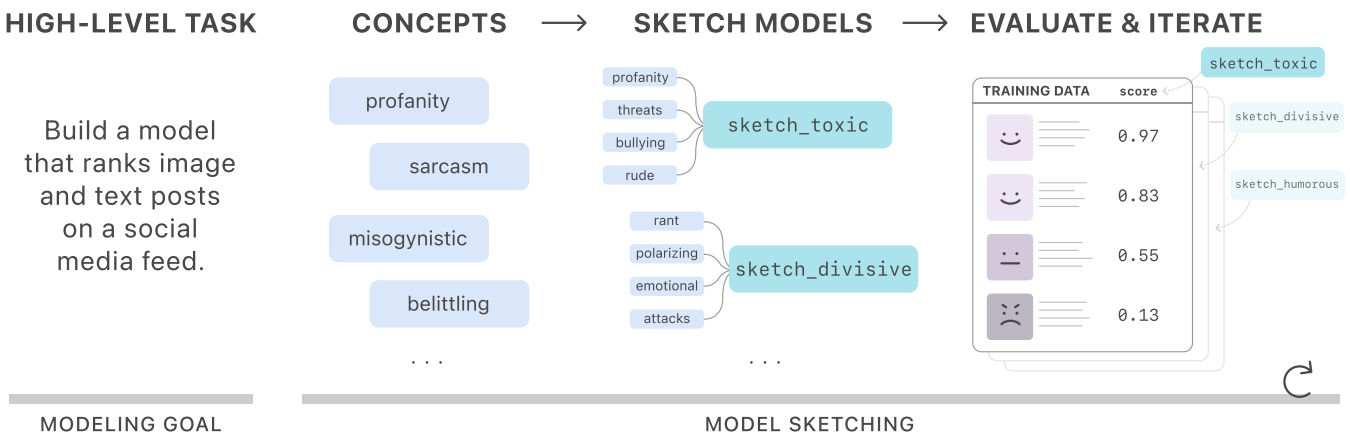

Figure 1: A conceptual diagram of the model sketching workflow. In just a few minutes, model sketching enables ML practitioners to author early-stage, functional models oriented around high-level concepts. We instantiate our approach in ModelSketchBook, a Python package for computational notebooks. *Concepts*: The user specifies concepts that are relevant to the decision-making task. Our system turns each concept into a functional model that can score unseen examples on that concept. *Sketch Models*: The user trains a sketch model, which aggregates concepts to predict ground-truth ratings for the overall modeling goal. *Evaluate & Iterate*: The user evaluates the sketch model's predictions on training data to make sense of its behavior and gaps. They then iterate on concepts and sketch models to explore alternate approaches and better align the model with their goals.

ABSTRACT

Machine learning practitioners often end up tunneling on low-level technical details like model architectures and performance metrics. Could early model development instead focus on high-level questions of which factors a model ought to pay attention to? Inspired by the practice of sketching in design, which distills ideas to their minimal representation, we introduce *model sketching*: a technical framework for iteratively and rapidly authoring functional approximations of a machine learning model's decision-making logic. Model sketching refocuses practitioner attention on composing high-level, human-understandable concepts that the model is expected to reason over (e.g., profanity, racism, or sarcasm in

*Work was performed while author was visiting Stanford HAI.

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

ACM ISBN 978-1-4503-9421-5/23/04…$15.00
https://doi.org/10.1145/3544548.3581290

a content moderation task) using zero-shot concept instantiation. In an evaluation with 17 ML practitioners, model sketching reframed thinking from implementation to higher-level exploration, prompted iteration on a broader range of model designs, and helped identify gaps in the problem formulation—all in a fraction of the time ordinarily required to build a model.

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; *Interactive systems and tools*; • **Computing methodologies** → *Artificial intelligence*; *Machine learning*.

ACM Reference Format:
Michelle S. Lam, Zixian Ma, Anne Li, Izequiel Freitas, Dakuo Wang, James A. Landay, and Michael S. Bernstein. 2023. Model Sketching: Centering Concepts in Early-Stage Machine Learning Model Design. 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, 24 pages. https://doi.org/10.1145/3544548.3581290

1 INTRODUCTION

Imagine that you are a machine learning practitioner tasked with creating a model to flag hateful image memes on a social media site. Once you dive into building and iterating on your model, the questions you are tackling quickly become ones like, "How would a transformer model perform if we train on these examples and use these hyperparameters?" Once you're in this mode, it is challenging to escape. How could we have helped your early model development work to instead stay focused on questions like, "What factors—such as violence, racism, sexism, and profanity—should our model pay attention to when determining what content is hateful?" Could machine learning tools help to center early efforts not on *how to execute a solution*, but on *what kinds of solutions to pursue* in the first place?

Current tools and approaches in machine learning (ML) promote a tunneling mindset that focuses attention on technical implementation [29]. These tools pull us into a cascade of low-level tasks—gathering a dataset, extracting features, planning out data pipelines, considering tradeoffs in model architectures, deciding on a performance metric—all to arrive at a first version of a model. Such tunneling is dangerous because it locks us into a single problem formulation, which often persists even as we iterate [45]. In contrast, it is the problem formulation itself that is the most important to iterate on early in the process, when high-level, underspecified decision-making goals such as "detect hateful posts" are translated into one of many concrete problem definitions [27, 45, 46]. The more value-laden and ethically challenging a problem area, the more ML practitioners need to take care during the model design process to get this translation right. Unfortunately, this translation work tends to get buried and de-emphasized [68].

To counter similar technical tunneling in their work, designers have long adopted a *sketching practice* that purposefully foregrounds fluid exploration of high-level design possibilities. Buxton defines sketching as any design process whose output is quick, plentiful, ambiguous, and minimal in detail [6]. Sketches distill an idea to its minimal representation, focusing attention on the most important aspects of a design direction. In doing so, sketches act as valuable cognitive tools for practitioners themselves to understand nascent designs [16]. What is the right minimal representation for ML model development? Interactive machine learning approaches (like Wekinator, Create ML, and Teachable Machine [2, 17, 21]) and prompt-based approaches (using Large Language Models [5, 63] and Stable Diffusion [53]) allow for rapid model development, but they are centered around inputs and outputs, largely abstracting away the decision-making logic in between. For problem formulation tasks, it's precisely that *decision-making logic* that we need to directly iterate on: what are the factors that our model should consider to make its decision? Today's tools do not yet allow ML practitioners to sketch a model's decision-making logic to rapidly answer these kinds of critical model design questions.

In this paper, we introduce *model sketching*, a technical framework that allows ML practitioners to create expressive, sketch-like versions of machine learning models. To allow for direct iteration on a model's core logic, we use *concepts*—the human-understandable ideas or factors that are relevant to a decision-making task—as the minimal representation for ML sketches. For example, concepts in a hateful image meme detection task could range from concrete factors such as "profanity," "violence," and "sexism" to more abstract factors such as "sarcasm" and "humor." While prior work has explored concept-oriented abstractions in ML [32, 33, 44, 52], a key difference in our approach is that we not only center concepts, but we also enable rapid instantiation and iteration of arbitrary concepts by leveraging zero-shot methods. This combination of concepts and flexible, rapid authoring unlocks new interactions where users can directly sketch model logic. We implement our model sketching framework in a Python package called *ModelSketchBook*, which leverages recent developments in zero-shot learning with pretrained models like GPT-3 and CLIP [5, 51] to allow users to turn open-ended concept descriptions into functioning building blocks. With our tool loaded in a computational notebook environment, users can specify a concept like "sexism" or "profanity" and provide several image or text examples. ModelSketchBook draws on pretrained models to score each input on how sexist it might be or how likely it is to include profanity, and then the user combines sets of concepts together to train a holistic *sketch model*. By interactively authoring concepts, incorporating them into sketch models, and inspecting resultant modeling outcomes, users can flexibly explore different modeling approaches and their ramifications.

We illustrate the generalizability of our model sketching framework through demonstration case studies and a field evaluation. We provide case studies ranging from political candidate research to creative inspiration to reviewer bias auditing (Figure 2). Then, in a field evaluation, we asked 17 ML practitioners to author sketches that would determine whether internet memes sourced from the Hateful Memes Challenge [31] should be removed from a social media site. This task sets a high bar for technical difficulty (involving interacting text and image modalities), involves subjective judgement (providing sufficient interpretive flexibility to create a broad model design space), and involves ethical tradeoffs (requiring a consideration of multiple competing viewpoints).

Model sketching successfully enabled participants to shift from a technical mindset to a higher-level, design-oriented mindset that reasoned about the concepts that the model should and should not capture. Before being introduced to model sketching, participants

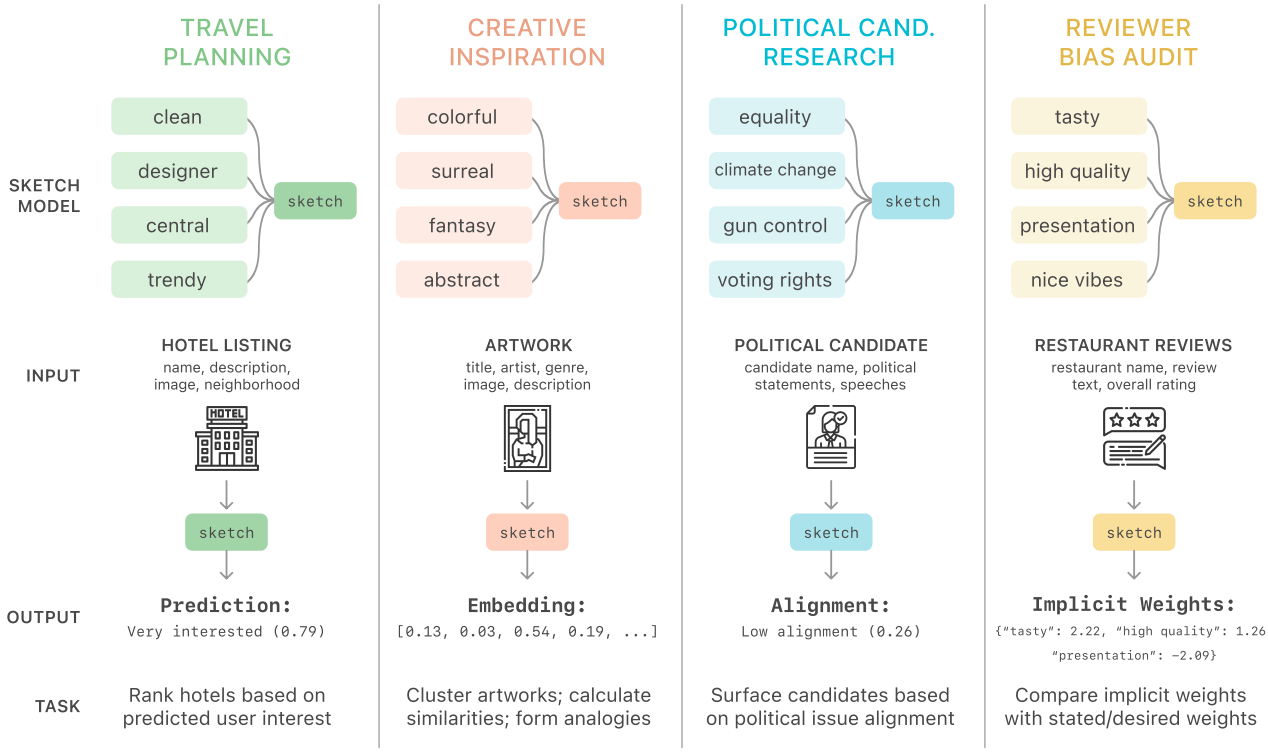

Figure 2: Model sketching case studies: (1) Travel Planning, a personalized recommendation task, (2) Creative Inspiration, an unsupervised clustering task, (3) Political Candidate Research, a personalized search task, and (4) Restaurant Reviews, a reviewer bias auditing task.

initially focused on technical details (e.g., model types like convolutional neural networks and transformers; performance metrics like F1 and AUC) when proposing a modeling plan for this task. After using ModelSketchBook, participants shifted to describe their approach in terms of higher-level design and problem formulation decisions. Participants iteratively experimented with concepts to address gaps exposed by their early sketch models: all participants branched out beyond their initial set of concepts, eventually covering concepts ranging from "racism" and "Islamophobia" to "sarcasm," "irony," and "controversial", that would have not arisen if the model were developed with traditional methods. With this expressive, lightweight interaction mode, participants turned their attention beyond modeling and identified gaps in the dataset, labeling method, and problem formulation; they noted factors such as inconsistent labels, class imbalance, lack of representation of certain types of harm, and subtleties of interaction between text and image semantics. While participants initially estimated that it would take from days to months to produce a model for this task, after using our tool for 20-30 minutes, the vast majority of participants (14 of 17) were satisfied with the models they created.

We view model sketching as an important step that moves beyond the mode of retroactively fixing broken models and towards one of helping ML practitioners to become better model designers from the start. To summarize, this paper makes the following contributions:

- **Model sketching**. We introduce a framework for authoring zero-shot concept building blocks that can be flexibly aggregated into composite models, allowing machine learning practitioners to rapidly iterate over concepts to sketch a model's decision-making logic.
- **The ModelSketchBook tool**. We instantiate our model sketching framework by introducing an open-source Python package that allows users to interactively author concepts and sketch models in computational notebooks.
- **An evaluation of model sketching with ML practitioners**. We perform a field evaluation with 17 machine learning practitioners who use our tool to author models for a challenging hateful meme detection task. We find that ML practitioners are successfully able to scope out from technical implementation details to higher-level decision-making logic. This cognitive shift helps ML practitioners to more broadly explore the model design space and to quickly discover gaps in the dataset, labeling method, and problem formulation.

2 RELATED WORK

We survey research on sketching and prototyping in design and HCI, the work practices of machine learning practitioners, and current approaches for early-stage model design, all of which motivate our model sketching approach.

2.1 Sketching and prototyping in design & HCI

The fields of design and HCI have long upheld the practice of sketching as a valuable tool for thought [6, 16]. A key trait of a successful sketching or prototyping tool is that it refocuses cognitive processes on high-level iteration, empowering the practitioner to engage in "design thinking rather than implementation tinkering" [23]. A sketch is similar to a prototype in that both are instantiations of a design concept, but a sketch serves a different purpose: to explore and raise questions about design approaches rather than to propose answers or solutions [6]. Thus, sketches are suitable for the early ideation stages of a design process while prototypes are intended for the later stages.

What characterizes a sketch? Buxton characterizes sketches as quick, plentiful, ambiguous, and minimal in detail [6]. The *quick* and *plentiful* attributes go hand in hand: methods with quick turnaround times enable practitioners to explore parallel design paths, an approach that supports more iterations, combats harmful design fixation, and leads to more diverse, higher-quality designs [13, 59]. Sketches are powerful tools because they allow a practitioner to quickly iterate to test dramatically different high-level ideas.

The *ambiguous* and *minimal detail* attributes are also closely tied: much of the value of a sketch lies in both the cognitive task of distilling an idea into a stylized, minimal representation [39] and in leaving enough ambiguity to suggest—to oneself and others—multiple interpretations (and thus multiple future directions) [20]. Early work in user interface design demonstrated that overly-detailed prototyping approaches detract from this valuable cognitive work [38] and do not substantially improve feedback quality [61]; in fact, simpler prototypes have been correlated with better design outcomes [65].

Finally, a sketch is *expressive*: it must provide enough flexibility and control to allow a practitioner to communicate their idea and iterate upon it [6, 38]. Traditional freehand sketches are highly expressive because designers can draw upon concrete or abstract visual and textual representations to hone their idea, allowing others to perceive both perceptual features and functional relations [58].

The machine learning domain presents distinct challenges for traditional prototyping methods due to AI's uncertain capabilities and complex outputs [66, 67]. However, we find that concept-based abstractions in particular hold promise for sketch-like approaches to ML model authoring. Prior work on concept evolution in data labeling demonstrated that many decision-making goals are malleable and subjective, but these goals can be clarified through an iterative process of concept elicitation and refinement [7, 34]. Researchers have even found that for real-life decision-makers such as judges, doctors, and managers, simple, statistically-derived rules based on a small number of human-interpretable features substantially outperform human experts [30]. Concepts map well to the way that humans reason over decision-making tasks, and methods that statistically aggregate concepts work surprisingly well, even beyond prototyping contexts.

2.2 ML practitioners: work practices and cognitive focus

In our work, we use the terms ML practitioner and model developer interchangeably to refer to users who perform work that primarily deals with designing, developing, and iterating on machine learning models [43, 68]. Recent literature frames the data science and ML project workflow as a multi-stage collaborative lifecycle where domain experts focus on high-level problem definition and results interpretation, while ML practitioners place their cognitive focus on low-level implementation details [50, 62, 68]. However, a major problem with the current workflow is that much of the translational work that ML practitioners perform to bridge between high-level goals and low-level implementation (i.e., via problem formulation, feature extraction, feature engineering) is rendered invisible [41, 43, 47, 68]. As a result, ML teams lose important knowledge about the nature and meaning of features and the assumptions and reasoning that underlie modeling decisions [25, 41], which is especially problematic if ML practitioners need to revisit those decisions to counter harmful model behavior. Similar patterns play out in statistical model authoring, which has prompted calls for high-level libraries to "[express] conceptual hypotheses to bootstrap model implementations" and bidirectional conceptual modeling to "[co-author] conceptual models and model implementations" [29]. Our work directly seeks to carry out this vision in the domain of ML model development by allowing ML practitioners to instantiate models using high-level concept-based abstractions that bridge to functional model implementations.

2.3 Early-stage machine learning model design

Work in interactive machine learning (IML) has long been interested in how we might engage humans in the model development process. In reaction to the slow and effortful process of building models, work in IML demonstrated that by enabling interactive model training and refinement, users can more rapidly develop models and keep them aligned with their goals [1, 15, 19, 48]. In most IML systems, interactions take the form of labeled examples and demonstrations, such as positive and negative image examples in CueFlik [18] and paired input gestures and output actions in Wekinator [17].

Building on IML, work on interactive machine teaching (IMT) emphasizes the metaphor of *teaching* to focus on the human's role as a teacher rather than just a label-provider [52, 57]. Through this metaphor, IMT highlights new challenges like the need for richer teaching strategies beyond data labeling. A wizard-of-oz study along this vein explored how users might teach knowledge to a model, and this work found that users decomposed knowledge in terms of concepts, relationships between concepts, and rules that combine and apply these concepts [44]. However, without actually instantiating and evaluating their concepts, users struggled to form a mental model of how the ML model works, which made it especially difficult for them to know how to articulate subtle, abstract forms of knowledge.

While model sketching shares many of the same motivations that underlie IML and IMT, it has slightly different goals—to explore a vague, unformulated model design space (problem formulation)

Paradigm	Emphasis	Paradigm shift	Strategy to improve ML models	Design requirements
Interactive Machine Learning	*interactive*	ML models don't need to exist in a vacuum—through human *interaction* and feedback, we can achieve user alignment and rapid model authoring	Better feedback loops	Speed
Interactive Machine Teaching	*teaching*	ML isn't just about learning—we need to focus on *teaching*: *who* is developing ML models and *how* they're sharing their knowledge	Better teaching	Expressivity
Model Sketching	*sketching*	ML models don't need to be high-fidelity—in fact, *low-fidelity, transient* models are better at helping us to explore the multiverse of model designs	Better design space exploration	Minimal detail, Speed, Expressivity

Table 1: A summary of key ideas put forward by Interactive Machine Learning and Interactive Machine Teaching, alongside those of Model Sketching. The three paradigms are complementary, but emphasize different perspectives and strategies.

rather than execute on a crisp modeling vision (model implementation). This means that in addition to the speed and expressivity required by IML and IMT, model sketching requires a *minimal representation of a model's decision-making logic* to facilitate effective design space exploration (Table 1). To support model sketching, we need to provide a mechanism for the user to directly iterate on the decision-making primitives that control a model's actions. Thus, we cannot use black-box methods that opaquely map from inputs to outputs, such as example-based IML approaches and prompt-based model authoring. Concepts are an ideal representation that allows for iteration on decision-making strategy and alignment with the ways that humans prefer to decompose and express their knowledge [44].

The metaphor of sketching also informs how we must instantiate these concept-based representations. Past work in IML and IMT has explored concept instantiation with examples, demonstrations, predefined features, and formal knowledge bases [4, 17, 18, 52]. However, given that sketches are meant to explore a broad model design space that is not known ahead of time, users need to have the flexibility to author arbitrary concepts, so we cannot rely on static features or knowledge bases. An added challenge of sketching is that there is a steep cost associated with diverting the user's attention to different tasks (e.g., switching to label examples, develop a feature pipeline, or train a new helper model), which risk pulling the user out of the sketching mindset and into technical tunneling. Thus, we build off of zero-shot prompt-based modeling approaches [28, 63] to instantiate concepts flexibly and rapidly without diverting attention.

3 MODEL SKETCHING

In this section, we describe our model sketching framework, which aids ML practitioners in moving beyond technical implementation to explore the conceptual model design space during early phases of ML model development.

3.1 The Model Sketching Framework

Model sketching aims to facilitate the translational, interpretational work that goes in between a high-level description and any particular low-level implementation [27, 45]. This work charts a path through the multitude of plausible strategies for any one high-level modeling goal, each of which ossifies a set of human values and ethical implications that are encoded into the model. We address the gap between high-level modeling goals and low-level implementation with the intermediary representation of *concepts*—human-understandable ideas or factors that are relevant to a decision-making task. By foregrounding this translational step between goals and their implementation, we aim to scaffold ML practitioners' reasoning and help them to more effectively sketch out and explore parallel solution paths.

3.1.1 Data prerequisites. We require only a small amount of data in table form (e.g., spreadsheet, CSV, Pandas Dataframe) where each row represents a single example and each column represents a data field. These data fields can contain text, image, or numeric data. We recommend that the dataset has at least 40-50 rows to aid sensemaking during model development, while also allowing for a split between training and test data. If the user provides ground-truth labels, our system can help with sensemaking around model accuracy; however, such labels are not required.

3.1.2 Concepts: Crafting functional building blocks of decision-making. Concepts are how we decompose the larger and more ambitious task of a sketch model into more manageable, human-understandable components. We turn concepts into functional components by mapping from a data example to a score that indicates its relevance to the specified concept, or how well it exemplifies that concept (Figure 3). For example, a concept for "colorful" might produce scores for a set of images indicating roughly how colorful they are; a concept for "profanity" might produce scores for a set of text comments that estimate how profane they are. Thus, concepts bear similarities to feature extractors, but they are not limited to lower-level, objective attributes, and they require much less time and effort to produce, as a new concept can be created in several seconds. Once created, a concept can be applied to any arbitrary example, seen or unseen.

We build upon advances in zero-shot modeling to instantiate concepts. Zero-shot modeling describes any machine learning modeling approach that does not require any training examples to perform a task. To create a concept, the user provides an input concept term (a brief word or phrase) and a data field (column name) from their dataset. Our framework then authors a zero-shot model that outputs a numeric score for that concept for every item in the dataset. For our concepts, we implement two methods: 1) zero-shot text classification via large language model (LLM) prompting using

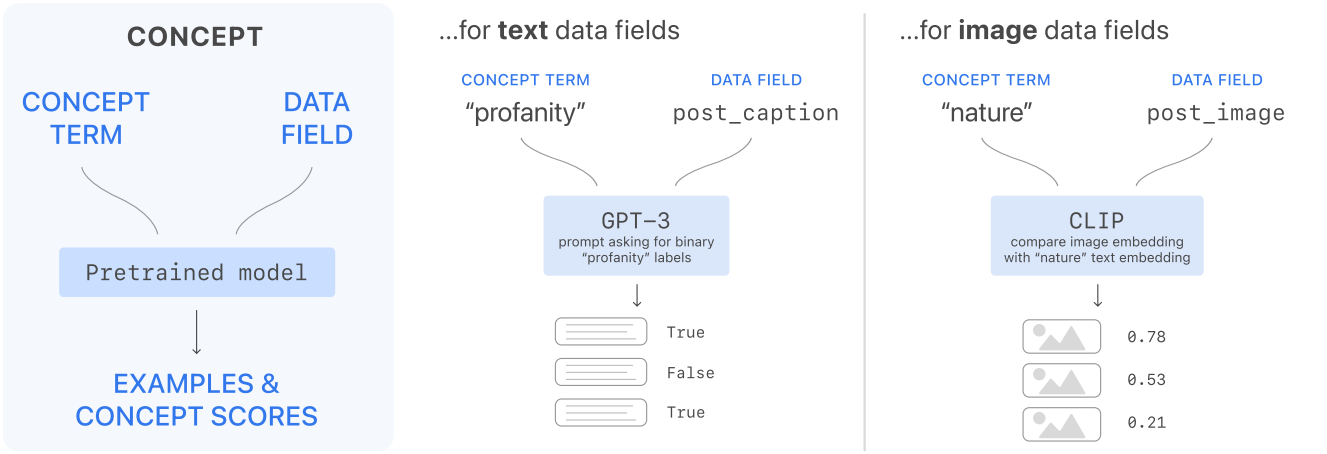

Figure 3: Users can turn human-understandable concepts into functional building blocks. A concept takes in a concept term and a data field as input. Then, a pretrained zero-shot model produces a score for each example indicating the extent to which the example relates to the specified concept. We adapt our underlying approach depending on the type of the data field. For text, we use GPT-3 prompts for binary label prediction. For images, we use CLIP embeddings for continuous score prediction.

GPT-3 and 2) zero-shot image classification via embedding similarity with a pre-trained image-to-text model using CLIP. We describe our technical approach in greater depth in Section 3.2.2. While we use these two zero-shot methods in our work, our model sketching framework is compatible with any current or future zero-shot modeling method.

3.1.3 Sketch models: Putting concepts together to envision decision-making outcomes. The second component of our framework involves authoring sketch models, lightweight ML models that aggregate the signals produced by concepts (Figure 4). A sketch model is designed to accomplish the same task as the full model that it is meant to emulate, so it expects the same inputs and produces the same class of outputs.

Once the user has authored a set of constituent concepts that are relevant to the decision-making task, they select which concepts they would like to combine into a sketch model. We provide an initial set of aggregators that can combine concept scores to produce a single sketch model score: linear regression, logistic regression, decision tree classification, random forest classification, and multi-layer perceptron (MLP) classification. Rather than require a complex deep learning model to integrate concepts, we build on prior research that demonstrated how simple linear models with small sets of human-interpretable features can achieve performance exceeding that of human decision-makers and rivaling that of complex prediction models [30]. However, we are not bound to this initial set of aggregators. We use the ground-truth labels provided in the training dataset to train these aggregators when a sketch model is created; all subsequent runs of the sketch model will evaluate the trained model on new input data. If the dataset does not have labels, then the user can opt to provide manual weights for the concepts using the linear regression aggregator.

With the sketch-model layer of abstraction, ML practitioners can focus their attention on experimenting with different combinations of concepts to create sketches that might capture different sets of values in decision-making. For example, when authoring a hate speech detection model, an ML practitioner may experiment with sketches that focus on race-related or gender-related hate speech, or approaches that explicitly account for intersectionality. They may try out sketches that take a relatively lenient stance by only adding a concept related to speech that incites violence, or they may experiment with sketches that take a more strict stance on profanity by adding concepts for swearing and slurs. While sketch models in their most stylized form are comprised entirely of concepts, sketch models are also compatible with more traditional features produced through custom feature extractors or existing ML models.

3.1.4 From sketch models to production models. While sketch models initially serve as tools for thinking and iterating during early model development phases, we envision that they could inform the design of full production models.

First, a key part of full-scale model design involves deciding how to *decompose the model* into smaller parts. A notable characteristic of successful production ML systems—consistent across use cases like feed ranking and content moderation and across platforms like Facebook, LinkedIn, and Reddit—is that they are composed of smaller, more targeted ML models whose outputs are synthesized into a single score, often via weighted linear combination [3, 12, 14]. In other words, the sub-models that ML practitioners produce in industry are more complex versions of our concepts, but they typically still boil down to human-understandable decision-making factors that will be synthesized into a single prediction via an aggregation process.

Second, a substantial part of model development (up to 80% by some estimates) is actually consumed by data cleaning and wrangling, and ML practitioners play an active role in shaping, curating, and designing data [43]. Model sketching could assist in refining *data requirements* by drawing out what kind of data is necessary for the task based on the concepts that the model must reason over. By testing out concepts on different data fields (e.g.,

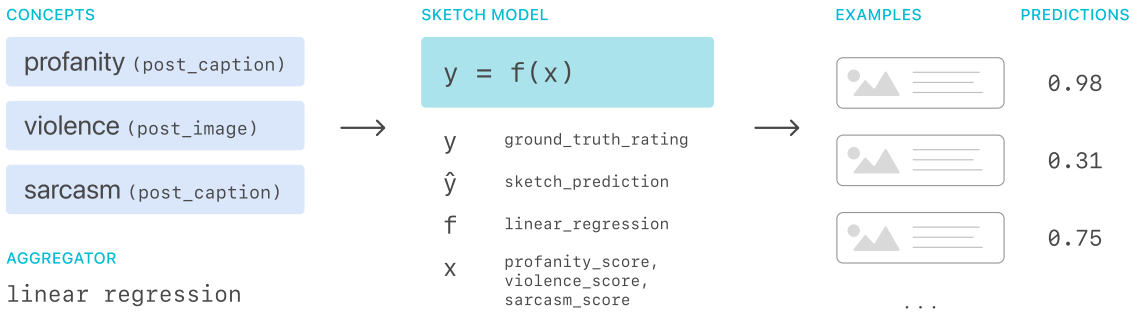

Figure 4: Sketch models combine concepts together to create an overall model for the decision-making task. First, the user chooses from among their authored concepts and decides on an aggregation method. We train a sketch model using the concept scores, the specified aggregator, and the user's ground-truth ratings (e.g., ratings of the hatefulness of posts). Then, the sketch model can produce predictions (e.g., scores estimating the hatefulness of posts) for any arbitrary example.

different text or image fields), ML practitioners might gain a quick understanding of what forms of data are most informative to serve different concepts and, ultimately, what data fields and what kinds of data examples should be required for the modeling task.

3.2 The ModelSketchBook tool

To instantiate our model sketching framework, we introduce the ModelSketchBook API,[1] a Python package that allows ML practitioners to create concepts and sketches of their own.

3.2.1 Environment and setup. Our API can be loaded in any Python environment, but it is optimized for use in a computational notebook such as Jupyter or Colab. We chose to tailor our API design to the notebook environment because this is where many ML practitioners already carry out much of their exploratory modeling work [54, 68].

```
import model_sketch_book as msb

# Set up the sketchbook
sb = msb.create_model_sketchbook(
    goal="Detect hateful memes on social media.",
    datasets={  # Pass in datasets to use
        "train": df_train,
        "test": df_test,
    },
    schema = {  # Specify the types of data fields
        "text": InputType.Text,
        "img_url": InputType.Image,
        "overall_rating": InputType.GroundTruth,
    },
)
```

After loading their dataset as a Pandas dataframe and importing the ModelSketchBook package in their notebook, the user sets up a sketchbook object. The sketchbook contains all of the datasets, concepts, and sketches that the user will author during their session, and it saves other metadata such as model predictions, performance metrics, and a history of concepts and sketches.

3.2.2 Creating concepts: under the hood. As mentioned previously, we specifically support text concepts powered by GPT-3 and image concepts powered by CLIP. To achieve zero-shot text classification, we author prompts that list a batch of examples for the model to classify and ask the model to decide on a binary label ("<concept term>" or "not <concept term>") for each provided example. Large language models like GPT-3 are designed to perform text completion on provided input prompts, so plausible completions provided by the model capture a rough sense of binary labels that might be reasonable based on the knowledge available in the model. We include an example prompt below.

```
Decide whether these comments are "motivational"
or "not motivational".
1. You can do it!!
2. Ugh, you're the worst.
3. I'm so tired...

Comment results:
```

The GPT-3 model will then return a text completion response similar to this that provides a label for the examples. We parse and post-process this GPT-3 response to retrieve binary labels.

```
1. motivational
2. not motivational
3. not motivational
```

To carry out zero-shot image classification, we use CLIP (Contrastive Language-Image Pre-Training) [51], a neural network trained on (image, text) pairs that allows images and text to be represented in a shared embedding space. Given a data field containing images and a concept term, we use CLIP to encode both the image and the concept term in that shared image-text embedding space. Then, we calculate and return the cosine similarity between the embedded image and concept term. Thus, the output score approximately captures the extent to which the given image relates to the specified concept term. We also allow users to normalize and re-calibrate these similarity scores to match their internal understanding of the concept, and we also allow users to apply custom thresholds to binarize continuous concept scores.

3.2.3 Creating concepts with the API. After setting up the sketchbook, users can freely author concepts. While we also provide raw API functions under the hood, for convenience, we surface UI elements via ipywidgets.[2] When the user makes a call to `create`

[1] https://github.com/StanfordHCI/ModelSketchBook

[2] http://ipywidgets.readthedocs.io

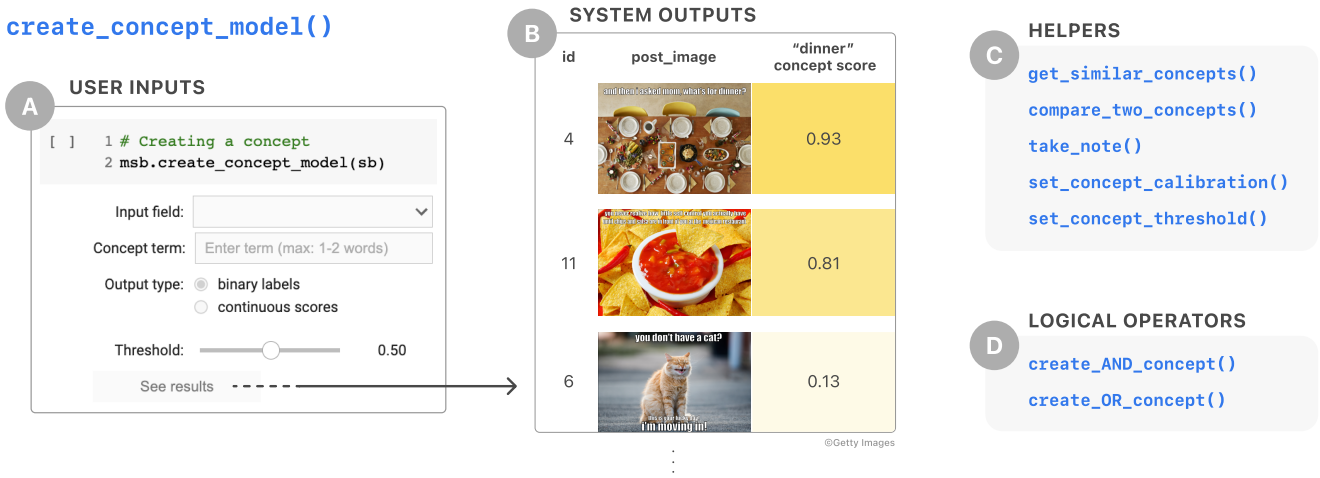

Figure 5: Creating concepts with ModelSketchBook in a computational notebook. (A) The `create_concept_model()` function accepts user inputs for the data input field, concept term, and other settings. (B) Then, the API outputs a dataframe displaying all training examples sorted by their concept score. (C) We provide helper functions for finetuning, making sense of existing concepts, and brainstorming new concepts. (D) We also support logical `AND` and `OR` operators to create compound concepts.

`_concept_model()`, a set of widgets will appear for them to specify the input field, a concept term, and other details like the output type of the concept score (Figure 5A).

Once the user enters in those details, they can execute the cell; after several seconds, the API returns a dataframe visualization that displays a sorted view of the examples in the training dataset along with their concept score (Figure 5B). The user can browse through the concept score results to determine whether the model's understanding of the concept sufficiently aligns with their own. If not, they can try out other concept terms or experiment with other input fields to work towards better alignment. The API also includes additional features for users to binarize continuous scores using a custom threshold value, auto-normalize continuous scores to a 0-1 score range, or custom-calibrate continuous scores based on a specified minimum and maximum score value.

We provide functions that assist users as they brainstorm and evaluate tradeoffs among different concepts. A brainstorming helper function surfaces synonyms or antonyms of a provided word to spur ideas when users seek to refine their concept. Meanwhile, a concept comparison helper function allows users to specify concept terms and compare their correlation with the input data and ground-truth ratings. To scaffold users' sensemaking process, we also provide a note-taking function that allows them to save scratch notes on concepts and sketches for future reference.

Finally, to allow for further control and refinement of concepts, we enable users to link existing concepts together with logical `AND` and `OR` operators to create compound concepts. With a call to `create_AND_concept()`, users can take the logical `AND` of any number of binary concepts (and can similarly link arbitrary concepts with an `OR` relation using the `create_OR_concept()` function). This grants users the ability to express more complex concepts, such as a single concept that can manifest in several ways or a multimodal concept that requires both image and text signals.

3.2.4 Creating sketches with the API. Once a user has authored several concepts, they can proceed to create a sketch model that aggregates those concepts by calling `create_sketch_model()`. This function displays widgets for the user to select the concepts and aggregation method (default: linear regression) for their sketch model (Figure 6A).

After executing the cell, the API creates the new sketch model and returns a dataframe visualization that displays all of the input fields of the training examples, all of the relevant concept scores, and the overall prediction produced by the sketch model (Figure 6B). This view also includes the ground-truth ratings and the difference between the sketch prediction and these ground-truth ratings. Users can choose to sort by various columns to aid their understanding of the model outputs. The sketch model output also displays traditional performance metrics for both regression and classification formulations of the task based on the ground-truth labels for the training examples; we display Mean Absolute Error (MAE), Classification Accuracy, F1 Score, Precision, and Recall.

Finally, the API provides functions that help the user to assess their sketch models. The `compare_sketches()` function allows the user to compare performance metrics for specified sketches by displaying the performance metrics side by side in a plot similar to Figure 11. The `test_sketch()` function allows users to test out sketch models on a labeled or unlabeled validation set (if one is provided) to get a sense for how well it generalizes to a broader set of examples. After the user selects a sketch model and a dataset, this function produces a dataframe similar to Figure 6B, only including the ground truth rating and error if present. Please see Appendix A for an overview of the three function calls required to instantiate a sketchbook, concept, and sketch model.

3.2.5 Implementation. The ModelSketchBook API was authored in Python. We use OpenAI's API for GPT-3 to perform zero-shot prompt-based text classification and use the `text-davinci-002`

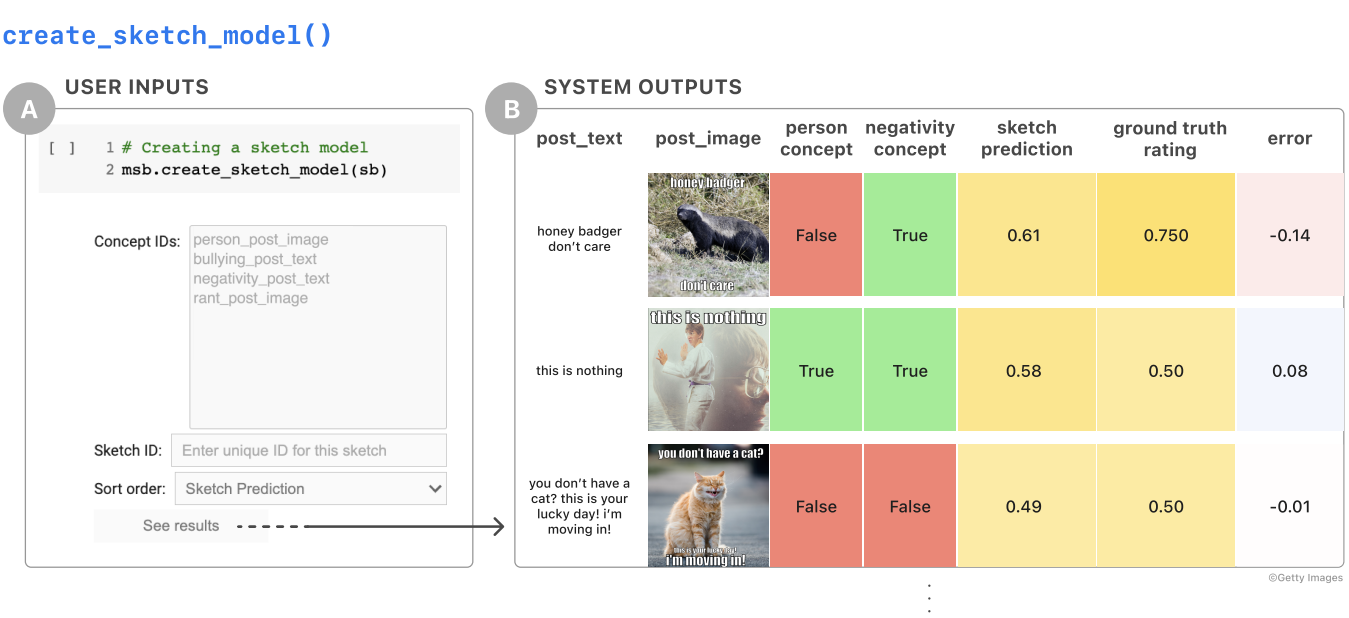

Figure 6: Authoring sketches with ModelSketchBook in a computational notebook. (A) The `create_sketch_model()` function accepts user inputs on the concepts to aggregate, with optional parameters to set the aggregation method and sort-column. (B) Then, the API outputs a dataframe displaying all training examples sorted by the specified column. This view shows all input fields, all concept scores, and the sketch model's prediction. It also displays the ground truth rating and the error between the sketch prediction and ground truth rating to aid error analysis.

model, the state-of-the-art GPT-3 variant at the time of development. We use OpenCLIP [26], an open-source version of CLIP, to perform zero-shot image-text alignment; we use a Vision Transformer (ViT) model (`ViT-B-32-quickgelu`) pre-trained on the `laion400m_e32` dataset. We use scikit-learn [49] model implementations for our sketch model aggregators. All training and inference takes place on the local machine on which the notebook is running (i.e., a user's own machine if they are running a local Juypter notebook, or a Google cloud server if they are running a Colab notebook). When loaded in a computational notebook environment, the input interface elements are rendered using the ipywidgets package. The output visualizations are rendered using Pandas dataframes with custom CSS styling.

4 DEMONSTRATION CASE STUDIES

To illustrate the breadth of machine learning tasks to which we can apply our model sketching approach, we present example case studies that we authored to demonstrate recommendation, unsupervised clustering, search, and bias audit applications (Figure 2; details in Appendix Table 5).

Travel planning—personalized recommendation. In our travel planning example, given a particular user, we aimed to recommend an Airbnb listing that would best match their preferences. Here, our sketches could capture key aspects of a user's taste in Airbnbs: based on images, we expressed preferred aesthetics ("clean," "bright," "designer") and red flags ("outdated," "ugly"); based on text descriptions, we conveyed desirable attributes of the neighborhood ("good vibes," "central," "trendy"). The sketching process helped us to discover gaps in the modeling setup such as the potential value of customer reviews to provide an objective assessment of factors like cleanliness and the importance of proximity and specific physical location-related information for the travel use case.

Creative inspiration—unsupervised embedding and clustering. Model sketching can also serve unsupervised learning tasks by allowing users to create customized sketch-like embedding spaces to perform item similarity, clustering, and analogical retrieval tasks. Building on a dataset of museum artwork, we authored sketch models for text and image fields related to bold visual style ("colorful," "surreal," "fantasy," "abstract"), social justice ("social justice themed," "race themed," "feminist," "queer themed"), and political movements ("politics themed," "revolution themed," "labor movement themed"). Then, we applied our sketch models to new artworks to retrieve their concept score-based "embeddings." We could then cluster artworks and gather creative inspiration by surfacing analogous works of art in disparate genres: by selecting an artwork in a painting genre and selecting a target genre of sculpture, we could retrieve analogous artworks from the sculpture genre that were most similar to the original artwork.

Political candidate research—personalized search. Our approach can also support search-related tasks. For this example, we assisted a user in researching political candidates before an election to discover which candidates were most aligned with them on the issues they cared about most. Drawing from third-party candidate information sources including platforms, political statements, and

speeches, we authored sketches that captured issues related to the environment ("climate change," "environment," "eco-friendly"), gun control ("gun control," "anti-gun), and other civil rights ("equal rights," "equality," "voting rights," "pro-choice"). Model sketching was especially beneficial for iteratively exploring nuanced candidate stances that weren't explicitly highlighted as core issues in standard voting guides.

Restaurant reviews—reviewer bias auditing. We can also apply model sketching for auditing tasks by working backwards from decision-making criteria to uncover latent weights from human or algorithmic decision-makers. For this example, we assisted a food reviewer in assessing their own potential biases. Based on the reviewer's self-reported decision-making factors (ordered by importance: taste, quality, presentation, vibes, service, and price), we created analogous concepts. Then, we authored sketch models to aggregate these concepts, using the reviewer's overall restaurant review ratings as ground truth. Our model inferred high positive coefficients for the "tasty" and "quality" concepts, which aligned with the reviewer's top two decision-making factors. However, the model inferred high negative coefficients for the "presentation" and "vibes" concepts, which corresponded to the reviewer's next most important factors. This revealed to the reviewer that these factors had less of a sway on their overall ratings than they intended.

5 EVALUATION

Our study aims to evaluate the impact of applying a model sketching approach to a realistic model authoring task with experienced machine learning practitioners. Since our goal is to shift the cognitive frame with which ML practitioners approach model authoring, our evaluation centers on these main research questions:

RQ1: *How does engaging in model sketching impact the cognitive focus of ML practitioners during model development?* Using this approach, do they successfully shift away from technical and implementation-oriented thinking and towards higher-level, concept-oriented thinking?

RQ2: *What are the outcomes of a model sketching workflow for ML models and ML practitioners?* When ML practitioners shift to higher-level thinking, what kinds of models do they produce? How do they anticipate modeling gaps and consider value tradeoffs in model design?

To address these research questions, we designed a field evaluation for participants with significant ML experience to use our ModelSketchBook API to author models that would detect hateful memes posted on a social platform.

5.1 Study design

5.1.1 Study format. Our evaluation consisted of an hour-long study session. Participants were sent a brief 10-minute labeling task to complete before the study session so that we could use their personal ground-truth ratings for the model-authoring task. We asked them to provide labels for 40 examples; 20 examples were used as a training set during the study session, and 20 were reserved as a test set. During the hour-long study session, the first 15-20 minutes were allocated to consent, Colab notebook setup, and a tutorial and video demo of the ModelSketchBook API. Then, the participant was given about 30 minutes to work on the model authoring task using the ModelSketchBook API in a Colab notebook. Finally, the last 10-15 minutes were split between a brief interview on their experience and a post-study survey questionnaire. Our full written response questions, survey questions, and interview scripts are included in Appendix Section D-F. We describe our qualitative analysis method in Appendix Section C.

5.1.2 The Hateful Memes detection task. Our dataset originated from the Hateful Memes Challenge [31] launched in 2020 by Meta AI to bring attention to the complex task of identifying multimodal hate speech. The dataset holds over 10,000 multimodal meme examples (combining text and images) based on real-world memes. The scoring scheme that we provided participants for their labeling task was: 0.00 = Benign (keep); 0.25 = Slightly problematic (may be questionable, but keep); 0.50 = Threshold (problematic enough to be removed); 0.75 = Problematic (remove); 1.00 = Very problematic (remove). Our formulation differs from the original Hateful Memes Challenge; we frame the labels around content removal rather than hate speech to orient the modeling task around the users' preferred decision-making logic rather than a centralized definition. Additionally, we filtered our dataset to examples that were labeled as "non-hateful" in the original dataset to reduce participants' exposure to unpleasant content and to skew towards more ambiguous, grey-area cases where users may make different value tradeoffs.

We selected this task for our user evaluation because we sought to evaluate model sketching in a difficult, realistic setting, and this task represented an open challenge that ML practitioners were actively working on. In addition, we wanted our task to be relatively subjective and not have clear right-or-wrong answers to provide participants a sufficiently broad model design space. Decisions around multimodal hate speech and content moderation commonly involve difficult value tradeoffs.

5.1.3 Pre- and Post-task written responses. Before receiving an introduction to model sketching, we asked participants to write out their modeling approach for the Hateful Memes task based on their current knowledge. After participants viewed the tutorial and demo, but before they started using the ModelSketchBook API, we asked them to brainstorm an initial set of at least three concepts that they might want to use in the task. At the end of the task period, we asked participants to once again describe their planned modeling approach for the Hateful Memes task, where they could assume that they had access to a tool like the ModelSketchBook API, but they did not have to use such a tool. We also asked participants to write down any learnings they had after engaging in this task, and we asked them to specify which sketch model they viewed as the "best sketch" among those they authored.

5.2 Participant recruitment

We sought participants who had significant experience with machine learning and IPython notebooks. We recruited participants by sending emails to university mailing lists for Computer Science and AI-related groups, and we sent recruitment materials to personal contacts working in the AI industry to share with their networks. We selected from among participants who answered either "rather

much" or "very much" (the two highest options on a 5-point Likert scale) to indicate their amount of experience with IPython notebooks and their experience with machine learning. In total, 17 users participated in our study (see Appendix Section G for participant demographics). We compensated participants with a $60 Amazon gift card for completing the hour-long study session. All studies were conducted remotely over video conferencing.

6 RESULTS

Using our ModelSketchBook API, participants successfully authored an average of 12.2 (SD=4.7) concepts and 4.1 (SD=2.0) different sketch models that built on those concepts for the Hateful Memes task. Figure 7 displays sample concepts and sketch models authored by participants, and Table 3 summarizes all participants' concepts.

Of the concepts authored in the study, 53.6% were text concepts, 32.4% were image concepts, and 14.0% were compound concepts involving a logical operator. Participants each authored on average 6.53 text concepts (SD=3.00), 4.19 image concepts (SD=2.14), and 2.07 compound concepts (SD=1.53). Of the compound concepts, 62.1% used an `AND` operator while 37.9% used an `OR` operator, and 79.3% combined only text concepts while the remaining 20.7% combined both text and image concepts. There were 136 distinct concepts authored by the study participants. On an individual level, participants each authored an average of 6.29 unique concepts (SD=3.77) that no other participants had authored.

We aimed for our modeling task to involve a high amount of subjectivity to provide a sufficiently broad model design space among participants. This was indeed the case: there were substantial differences in how our participants chose to label the provided dataset. Across participants and data examples, the mean standard deviation in labels (on a 0-1 scale) was 0.23. For the binarized versions of those same scores, this meant that 75% of the examples had at least one participant who disagreed with the majority label, and 37.5% of the examples had a quarter or more of the participants who disagreed with the majority label.

6.1 Changes in the cognitive process of model authoring

Addressing RQ1, model sketching aided ML practitioners in shifting their thinking from technical, implementation-oriented factors toward higher-level, conceptual, design-oriented factors of model development.

6.1.1 Changes in planned modeling approach. In their pre-task modeling plans, participants tended to focus on implementation details such as specific modeling methods or architectures (e.g., transformer models, convolutional neural networks, supervised models, linear classifiers), ML conventions like train-test splits and performance metrics, and data-oriented plans for feature extraction (summarized in Table 2). For example, P13 described their approach as "two types of models: [one] sentiment based and another that is a simple CNN, and then combine the two models with a simple fully connected layer," and P8 planned to "create test/val/train sets with equal distributions of data" and then "train some model and inspect how it performs on a held out test set."

However, after exposure to model sketching, participants described that they would spend time on brainstorming and curating

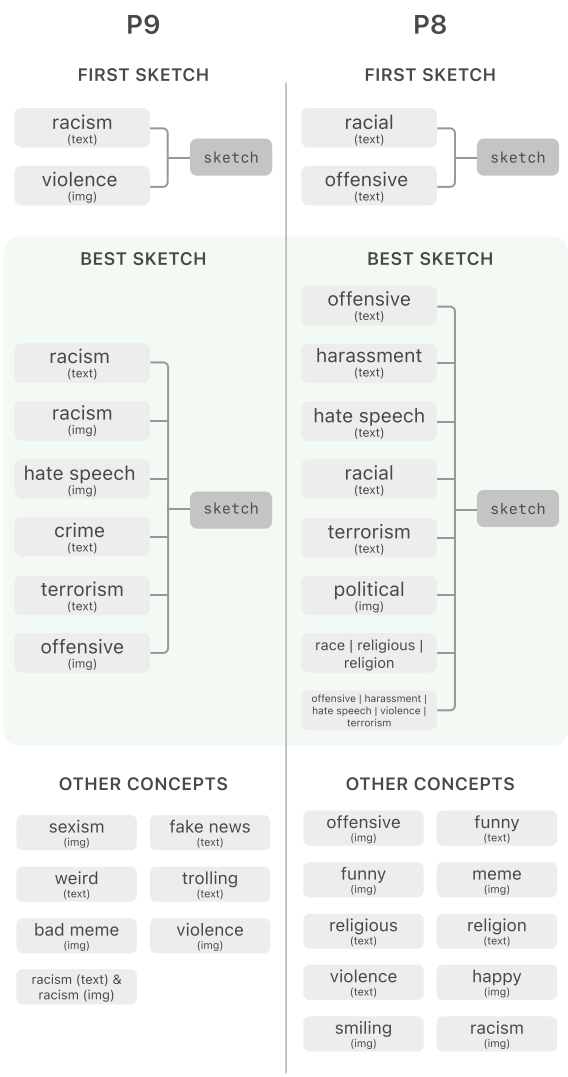

Figure 7: Examples of participant sketches. For illustrative purposes, we display the first sketch, best sketch, and other concepts for two study participants. Participants explored a broad range of concepts and expanded from their initial sketch models.

concepts relevant to the decision-making task and that they would iterate on their concepts by inspecting sketch model outputs. P15 described that "Spending a few days tinkering with concepts could produce a fairly effective model that would be easier to debug." Participants also branched out to mention that they would like to consider different subclasses of harm and described how they might combine a concept-based approach with more traditional modeling approaches. For example, while P8 still preferred to use a more complex, non-linear model, they wished to use concepts from ModelSketchBook to augment their datapoints. Participants also described changes to their mode of thinking during model sketching, which we qualitatively summarize in Appendix Section I.

Study Phase	Modeling Plan Theme	Quotes
Before model sketching	Specific model architectures or techniques	"I would use a convolutional neural network model [...] probably use transformers in conjunction with CNNs to track long-term dependencies and help contextualize the text data." (P1), "Two types of models: sentiment based and another that is a simple CNN, and then combine the two models with a simple fully connected layer." (P13)
	Evaluating on training and validation sets	"Ensure that the dataset is divided into train, test and validation slices. Experiment with several modeling techniques while evaluating on the validation set." (P3)
	Standard performance metrics	"Measure various metrics (accuracy, bias, variance etc)." (P1), "Figure out the right set of metrics for evaluating any model that is trained. Choices could be accuracy, F1 scores and other metrics such as AUC or AuPR depending on relative label weights." (P3)
	Building low-level features stemming from data	"Can use features from the posts and user interactions. Can use general user information along with the post information." (P4), "Build features representing intuitions for things relevant to removal; like topic metadata, language choice, likes/dislikes/controversiality." (P5)
After model sketching	Spending time brainstorming and curating concepts	"I would likely lean into feature engineering carefully to have an interpretable model. Spending a few days tinkering with concepts could produce a fairly effective model that would be easier to debug." (P15)
	Iterating on concepts by inspecting sketch model outputs	"[I would] see mislabelled (and correctly labeled) examples from the aggregator and brainstorm more specific features that might account for that." (P5)
	Authoring concepts informed by different categories of harm	"I think filtering on hate (race/gender/religion) speech and political speech would be good first steps. There are some examples of graphic/violent posts that might not be captured by those categories, so I would add more filters for those specifically." (P2)
	Combining a concept-based approach with traditional approaches	"Also can use these labels as annotations to the datapoints for a more complex model (nonlinear). Seems like having the original data is still good so wouldn't rely on the CLIP/GPT annotations alone." (P8), "I will probably still collect data manually and then instead of using a pretrained model and build the pipeline from scratch, I will use the ModelSketchBook API." (P11)

Table 2: Participants' modeling plans before and after the study. Participants shifted from more technical, implementation-related plans to more procedural, value-oriented plans and chose to adopt model sketching in their future modeling plans.

6.1.2 Concept evolution. One window into a participant's mindset is the way that their concepts evolve over time while using ModelSketchBook. We observed that on average, participants engaged in 8.5 (SD=2.7) instances of *concept innovation*, which we define as the number of distinct concept ideas; on average 5.0 (SD=2.8) of those concept innovations were *not* in the participant's original brainstormed list. Concept evolution often stemmed from participants discovering errors in their sketch models: P7 noted that a sketch model combining racism and sex concepts ran into false positives with humorous posts, which inspired an exploration of humor-related concepts. These progressions often built on each other: P4 initially started with a profanity concept that worked well, so they then brainstormed other dimensions of hate and decided to explore racism-related concepts; they then discovered that race and ethnicity-related hate in this dataset was often targeted at Muslims, so they got the idea to explore concepts related to Islamophobia. Participants on average displayed 3.2 (SD=2.3) instances of *concept execution*, which we define as the number of concepts authored that were continuations of the same concept idea (for example, switching from an image-based racism concept to a text-based racism concept, or changing from a concept term of "racism" to one of "race"). Participants more frequently changed modalities, but rarely modified concept terms. Then, participants underwent *concept abandonment* for an average of 4.2 (SD=2.6) concept ideas; we define this phenomenon as the number of concept ideas that were not ultimately used in the user's self-designated "best" sketch model. This abandonment is also a valuable part of sketching—to identify and discard paths that appear to be dead-ends. For example, P14 tried a "harmful" image-based concept, but found that this surfaced noisy results; they concluded that this was because the concept was too vague and didn't closely relate to the images in the dataset, and they decided to explore more grounded, specific concepts for images. Thus, participants appear to engage in a variety of different generative, iterative, and reflective modes as they author concepts.

6.2 Modeling outcomes

Regarding RQ2, we observed that the high-level cognitive shift in focus during model development brought about concrete benefits for the models themselves—from the breadth of concepts that sketch models explored to the gaps that participants uncovered in the modeling setup.

6.2.1 Identifying modeling gaps. Model sketching helped participants to gain insights beyond just the model itself. Participants uncovered major categories of harm that were not sufficiently represented in the provided sample of the dataset, such as homophobia and transphobia, and they noted that memes in the dataset appeared unrealistic compared to current online memes, so models built on this data may fail in practice due to distribution shift. Participants also identified limitations of the labeling approach such as class imbalance and inconsistent ratings. They gained intuitions about the task itself: participants discovered that for these memes, text was often more meaningful since the images were usually quite benign, and they started to form a sense of cases where multimodal

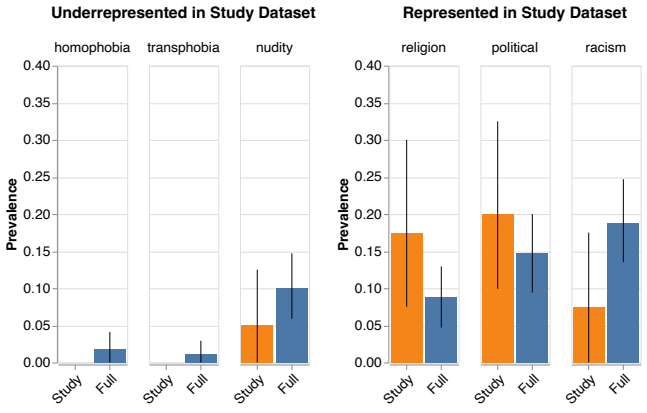

Figure 8: Concepts that participants noted as underrepresented based on their sketching process with the study dataset (left) were generally underrepresented in the full dataset. Already-prevalent concepts (right) maintained substantial representation in the full dataset.

approaches may reap more benefits. See a more detailed summary in Appendix Section H.

After the study concluded, we sought quantitative evidence to support the modeling gaps that participants raised.

Dataset representativity: First, we explored whether the concepts that participants claimed were insufficiently represented in their 40-example dataset were indeed underrepresented in the full 8.5k-example training dataset. Since the full dataset was not annotated for these concepts, a member of our team performed manual labeling for the three participant-identified concepts ("homophobia," "transphobia," and "nudity") for both the 40-example study dataset and a random 2% sample of the full training set (170 examples). As a point of comparison, we additionally performed manual annotation for three participant-authored, higher prevalence concepts—"religion," "political," and "racism"—which had 17.5%, 20.0%, and 7.5% prevalence in the study dataset, respectively. We found that indeed homophobia, transphobia, and nudity, which had 0%, 0%, and 5% prevalence in the study dataset, also appear underrepresented in the full training dataset with 1.8%, 1.2%, and 10% prevalence (Figure 8). Overall, we see that concept representativity in the small dataset used for model sketching can provide helpful warning signals on representativity in the full dataset.

Class imbalance: Then, we characterized the extent of class imbalance among participants' labels. During the study, four participants raised the issue of class imbalance; all of these participants indeed had a high level of imbalance, and all were skewed toward negative labels (mean=79.3% negative). Two of these participants displayed the highest class imbalance among all participants (with 92.5% and 87.5% negative labels). In addition, outside of the participants who noted the issue, three other participants also had more than 75% negative labels. Thus, it appears that indeed class imbalance was a common challenge for this task.

We were encouraged to find that participants were able to rapidly identify potential modeling gaps before gathering a large-scale dataset or implementing a full model.

Theme	Sample Participant Concepts	Count
Ethnicity, Race	racism, racist, race	34
Danger, Violence	violence, terrorism, harm	28
Emotion	rant, scary, anger	27
Discrimination	hate, hate speech, discrimination	22
Politics	political, politics, fake news	16
Offensive	offensive, vulgar, obscene	15
Religion	religion, religious, muslim	14
Gender	sexist, sexism, gender	10
Sexual Content	nudity, sexual, sex	9
Humor	funny, humor, joke	8
Trolling	trolling, meme, bad meme	7
Intersectionality	color & gender, islamophobia | sexism	7

Table 3: Summary of participant concepts for themes covered by 3 or more participants. We display the most common concept terms for each theme and the total number of participant concepts that fell within the theme.

6.2.2 Concept diversity. Participants experimented with a broad variety of concept terms—ranging from concepts related to race and ethnicity to concepts related to politics, religion, and violence (summarized by theme in Table 3). Notably, all participants authored concepts that they had not considered in their initial brainstorm; the process of authoring concepts and sketches helped them to discover important and relevant concepts to incorporate. Mapping participants' individual concepts to the 12 common themes that we identified, we found that participants created concepts that spanned on average 5.65 of these themes (SD=1.90). In the reverse direction, each theme was covered by an average of 8.0 participants (SD=3.10). We thus see that participants were able to explore a diverse set of concepts in their sketching sessions, and many concept themes were investigated from a variety of angles by different participants (Figure 9).

6.2.3 Time required. Finally, we note that participants were able to achieve these positive modeling outcomes in a fraction of the time that would be required to author a full model for this task. Participants' time estimates for executing on their pre-task modeling plans for the Hateful Memes task varied drastically from half a day to several weeks or months. However, even considering the most optimistic estimate of several hours (provided by 3 of 17 participants), our model sketching approach was able to deliver functional model prototypes in less than 30 minutes. Beyond the speed of model sketching, we document participant experience outcomes in detail in Appendix Section J.

6.3 Benchmarking concept performance

While we found in our case study demonstrations and field evaluation that concepts worked sufficiently well to support users' sketch modeling process, we sought to characterize the accuracy of our zero-shot concept scores compared to human annotations. We selected the most frequently-instantiated concepts among our study participants, which had been independently used by four or more participants. These included four text-based concepts ("religion," "political," "racism," and "violence") and three image-based concepts

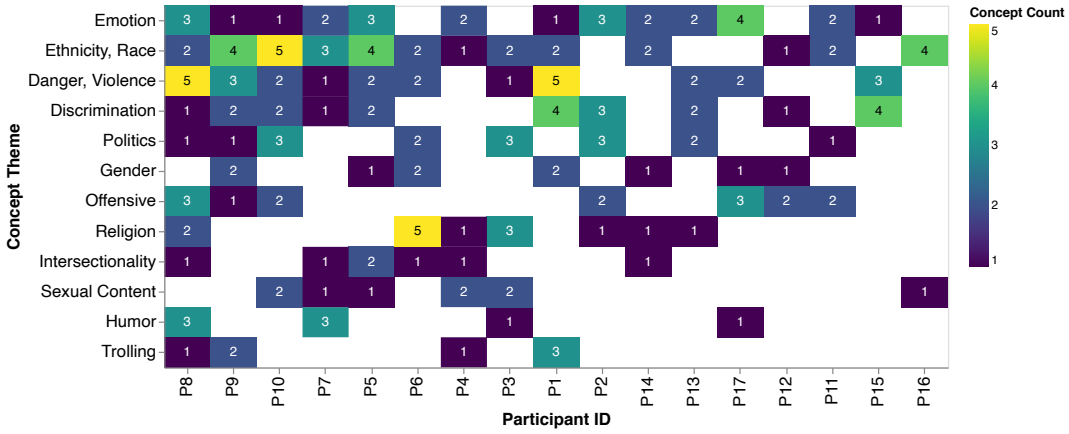

Figure 9: A heatmap matrix mapping from participants (x-axis) to concept themes (y-axis) depicts substantial spread between participants and themes, indicating that participants each explored diverse sets of concepts. Concept themes are sorted from top to bottom by the unique participant count, and participants are sorted from left to right by the unique concept theme count.

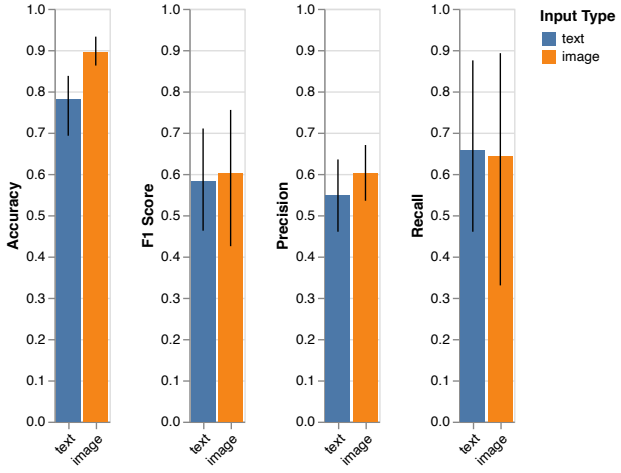

Figure 10: Performance for participants' text and image concepts is moderately high based on manual annotations.

	Concept Type	Text	Image
Annotation metrics	**Mean Prevalence**	19.4%	13.3%
	Fleiss' Kappa	0.68	0.60
Concept performance	**Accuracy**	0.78	0.90
	F1 Score	0.58	0.60
	Recall	0.66	0.64
	Precision	0.55	0.60

Table 4: Performance estimates were based on concepts that were sufficiently prevalent in the dataset and that displayed substantial inter-rater agreement.

("racism," "nudity," and "violence"). Four members of our team independently provided ratings for these concepts on all 40 items shown to participants in our study. Based on these manual labels, the prevalence for the image-based concepts was too low to provide a reliable estimate, with a mean of 4.2% prevalence (including 0 examples for the "nudity" concept) in participants' datasets.[3] We thus repeated the same procedure with the next most frequently-instantiated image-based concepts among our study participants ("muslim," "historical," "political," and "animal").

We observe comparable levels of performance for both text and image concepts, as summarized in Figure 10 and Table 4. Both sets of concepts displayed substantial levels of inter-rater agreement as measured by Fleiss' Kappa and had sufficient prevalence levels based on these manual labels. For the text-based concepts, we found a mean accuracy of 0.78 (SD=0.09) and a mean F1 score of 0.58 (SD=0.14), with a higher recall than precision (Recall=0.66, Precision=0.55). We found that for image-based concepts, there was a mean accuracy of 0.90 (SD=0.04) and a mean F1 score of 0.60 (SD=0.20), with higher recall than precision again (Recall=0.64, Precision=0.60). Thus, for an approach that requires no data collection, concept labeling, or training time, the zero-shot concept scores provide users with a reasonable starting point for further iteration and refinement.

6.4 Benchmarking sketch model performance

While the goal of model sketching is not to produce full-scale models, we sought to understand whether sketch models achieve reasonable functionality relative to full-scale models.

6.4.1 Study participants' sketch models. First, we investigated the sketch models authored by our study participants. We compared all participants' self-determined "best sketch model" against two

[3]This relates to the same issues that participants raised in Section 6.2.1 about the need for greater concept representation in the Hateful Memes dataset.

kinds of baselines: 1) *first sketch*, the very first sketch that a participant authored with the ModelSketchBook API and 2) *zero-shot*, a baseline that used a single GPT-3 prompt to predict whether each example is "hateful" or not. For each model variant, performance was measured against the participant's own ground truth labels on their 20-example test set. We found that participants' best sketches outperformed both baselines on classification metrics (F1, precision, recall), but the best sketches achieved slightly lower performance than the baselines on the regression version of the task, as measured by Mean Absolute Error (MAE) (Figure 11). Looking back at performance on the 20-example training set upon which users were actively iterating, participants' best sketches more strongly outperformed the first sketch and zero-shot baselines on all metrics.

These results suggest that participants are able to iteratively improve their sketch models to achieve higher alignment with their modeling goals and that sketch models may generalize better for classification tasks than for fine-grained regression tasks. Given that sketch model performance metrics declined between the training set and test set, participants may have overfit to the training dataset during their sketch model development. Meanwhile, the zero-shot baseline displayed a generally worse—but consistent—level of performance between the train and test set as expected, precisely because this model was given no opportunity to overfit to the training set. These two extremes fall in line with classic bias-variance tradeoffs in machine learning. Our sketch model results mirror prior literature on the common challenge of overfitting tendencies in IML [11, 64], especially given that we did not take steps to explicitly steer users away from overfitting behavior. Lastly, we note that for this task, the classification metrics were relatively low overall due to class imbalance as discussed in Section 6.2.1. Many participants had substantial skew in labels toward the negative class ("non-hateful") and thus ended up producing models that always predicted the negative class (behavior which results in scores of 0.0 for recall, precision, and F1).

6.4.2 Investigating the impact of task difficulty on sketch model performance. Then, we sought to further investigate the performance of sketch models in cases without such strong class imbalance and with many more than 20 training examples. We prepared three additional tasks: an IMDB movie review sentiment analysis task (easiest: per-example ratings and balanced classes) [60]; a comment toxicity task (harder: per-annotator ratings and class skew) [35]; and the original Hateful Memes task (hardest: balanced classes, but nuanced and multimodal task) [31]. For each of these, we compared sketch models against published state-of-the-art (SOTA) models, using the same headline performance metric and train/test datasets as were used by the SOTA model. Please see Appendix Section K for additional details on the benchmarks and our sketch models. For the IMDB task, the SOTA model reported 0.93 accuracy and our sketch model achieved 0.91 accuracy, only slightly lower. On the comment toxicity task (which used a 0-4 score range), the SOTA model reported 0.90 MAE while our sketch model achieved an MAE of 1.39. On the Hateful Memes task, the SOTA model reported 0.845 AUROC, and 0.714 AUROC was the SOTA result when the challenge was first released; our sketch model achieved 0.584 AUROC. Thus, model sketching demonstrates strong or reasonable performance on the easy and medium difficulty tasks, but displays weaker performance results for the third, significantly harder task.

6.5 Failure modes

While our model sketching approach achieved its aims of shifting model developers' cognitive frame and presented a number of benefits for the models they developed, we observed several failure modes. First, though participants had some promising concept ideas, they were not always able to actualize them with our zero-shot model-based approach. Participants often relied heavily on the results of the first concept term that they attempted. If that failed to return satisfactory results, they often abandoned the concept idea since they assumed that the model was not sophisticated enough to understand that concept. Participants would benefit from scaffolding to understand whether the model lacks an understanding of a given concept or just requires different terminology.

Second, we found that some participants heavily tailored their concept terms based on what they thought zero-shot models would be able to understand, especially using concrete, specific concept terms. We observed that several of these participants eventually branched out and found that broader terms could work well. However, the downside still remains that since our concepts rely on pretrained models, these models can be limiting factors (further discussed in Section 7.2.1). Users might constrain their thinking to areas where the pretrained model already excels rather than thinking more creatively about what decisionmaking factors may be valuable.

Finally, we noticed that because our approach emphasized a "small data" approach to help users stay focused, some participants lost sight of the overall goal to build a robust, general model. Participants sometimes fell into overfitting approaches by scrutinizing cases where their sketch model made errors and patching errors with very specific concepts that were not relevant to the larger decision-making task. Participants would benefit from tools that would help them to assess the generalizability of their sketch models. We believe that a curated dataset is worth this tradeoff in the sketching context because it encourages ML practitioners to avoid detailed data analysis and instead focus on broad-strokes concepts.

7 DISCUSSION

We have introduced model sketching, instantiated the approach in a tool designed for ML practitioners, and demonstrated that the approach successfully redirects attention to the high-level model design questions that are critical to get right early on. Here, we discuss the opportunities afforded by model sketching as well as limitations and areas for future work.

7.1 Opportunities for model sketching

We envision several exciting implications for bringing a sketching practice to machine learning.

7.1.1 Expanding the purview of sketch-inspired ML model development approaches. While our work supports compound concepts with logical operators to add multimodality and more nuanced concepts, these are more cumbersome to author and may enforce too much rule-based rigidity. Future work might explore multiple

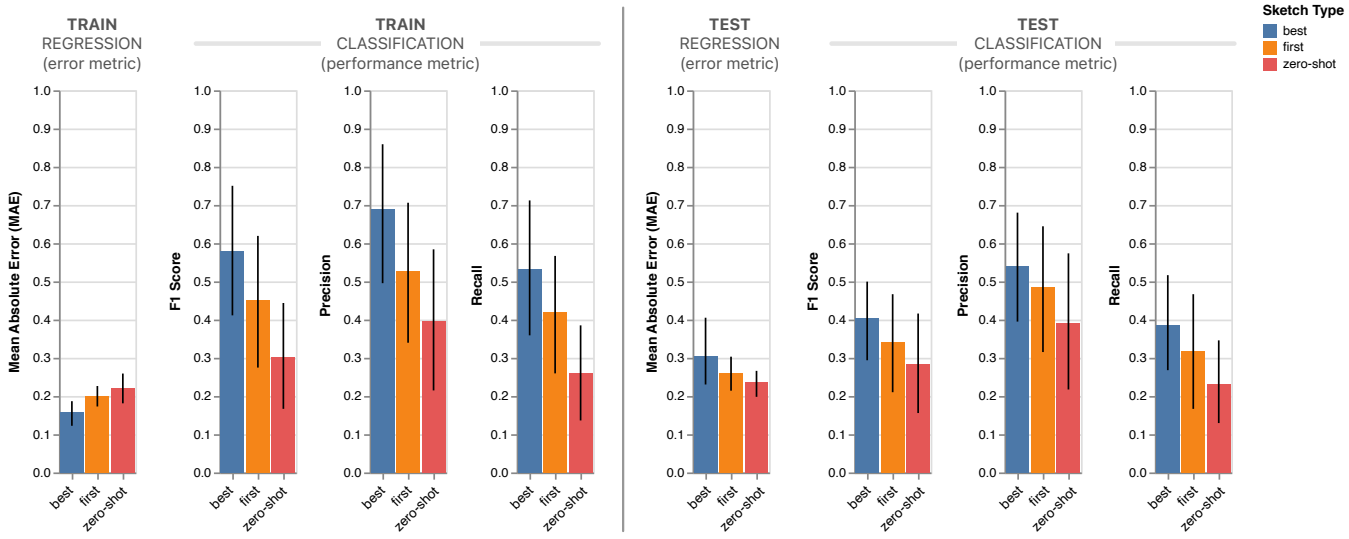

Figure 11: For the training set (left), we observe consistently better performance (MAE, F1 score, precision, and recall) for participants' self-designated "best" sketches compared to baselines of their first sketch and a zero-shot model. For the test set (right), we observe better classification performance (F1 score, precision, and recall) for the "best" sketch, but higher MAE.

nested levels of model sketching to express complex concepts. For example, a concept like *racism* is extremely multifaceted; racism has many forms and can be experienced in many different ways. Thus, rather than just accept a generic definition as given, a model designer must take care to specify the nuanced notion of racism that their particular problem formulation and model deployment context requires. Through nested sketch models, ML practitioners could iteratively "zoom in" to refine concepts and subconcepts as needed.

7.1.2 Bringing sketching to model-adjacent tasks. Participants in our evaluation uncovered gaps not just in their models, but in the larger modeling ecosystem of problem formulation, data, and labeling methods. Data work in particular tends to be neglected and undervalued relative to modeling work [24, 55]. The model sketching approach could be extended to assist with some of these tasks. For example, in data annotation tasks, a major challenge lies in iteratively developing a refined labeling policy [34]. Model sketching could iteratively simulate the results of different labeling policies, exposing areas where further concept refinement is needed *before* ML practitioners launch costly human labeling efforts.

7.1.3 Empowering non-technical users to participate in ML prototyping. Currently, stakeholders who lack ML experience (whether end users, domain experts, designers, or product managers) have little ability to take ownership of ML model design decisions and are largely limited to the information that ML experts convey to them [37, 50]. Model sketching could enable non-experts in ML to build functional sketch models that incorporate their domain expertise, a highly valuable contribution given that ML practitioners typically lack expertise in particular deployment domains. Several participants in our user evaluation explicitly brought up that model sketching would be very useful for non-technical users.

7.2 Limitations and future work

7.2.1 Expanding beyond the constraints of pretrained models and zero-shot modeling. One limitation of our technical approach based on zero-shot modeling using pretrained models is that these models may not be flexible enough to support a broad range of modeling tasks. Given that the GPT-3 and CLIP models were trained on Internet text and image data, these models have a limited view of the world and cannot provide high-quality concept scores for domains that are not well represented in this training data. Future work might explore the efficacy of other specialized pretrained models or alternative zero-shot modeling approaches that could overcome these gaps.

While we restricted ourselves to zero-shot concept instantiation to preserve a sketch-like interaction, further work might take hybrid approaches that allow for example-based concept specification. As we found with our concept generalization benchmarking, zero-shot concepts achieve reasonable performance on average for the purpose of sketching, but they are not consistently accurate and may benefit from more manual control. Through few-shot modeling or fine-tuning on pretrained models, users might be able to refine their concepts with annotated examples in cases where that refinement is worth the additional time investment.

7.2.2 Ensuring proper use and interpretation of sketch models. We must take care to communicate to users the intended purpose of our tool. Since our approach bridges between human-understandable concepts and model predictions, ill-informed users might attempt to use our method for interpretability or explainability: purposes that our tool was not designed to support. Trading off accuracy for speed and flexibility, our model sketching approach was designed to assist with early-stage model design exploration. Meanwhile, an interpretability use case requires a much higher bar of concept

and model accuracy. A model sketching tool must make these distinctions clear. By limiting the dataset size and orienting our tool around novel model authoring rather than post-hoc analysis of production models, we also steer users away from these unsupported use cases.

Additionally, as with any technology that builds on top of large language models and other pretrained models, sketch models will be subject to biases embedded in these models [36, 56]. It is crucial that ML practitioners are aware of these risks, so we encourage others who adopt model sketching to make clear to users that these underlying models can display harmful biases that may manifest in their sketch models. Though these biases in themselves are negative, one positive side effect is that encountering them early in the model development process forces ML practitioners to consider how the same kinds of biases might impact their full-scale models. Early encounters with such failures may prompt model developers to prioritize fairness-related work, and our approach allows them to experiment with different strategies—whether with the modeling approach, upstream data curation, or downstream score processing and bias checks—that might mitigate the harmful biases they encounter.

7.2.3 Bridging from sketches to full-scale models. While sketch models are early explorations rather than final versions of a model, they still need to provide useful signal on a full-scale model's behavior. As detailed in Section 3.1.4, sketch models might inform the design of production models by guiding decisions on how to decompose large modeling tasks and refine data requirements. Our technical benchmark against state-of-the-art models suggests that model sketching can achieve substantial performance on some tasks, but may struggle to generalize on more challenging tasks. The model sketching process still provides value in facilitating reflection and iteration on high-level modeling approaches even if performance metrics hit a ceiling. However, important areas of future work will be to support the bridging work between a model sketch and its production deployment and to provide stronger guarantees on the mapping between sketch and full-scale performance metrics.

8 CONCLUSION

In a space dominated by high-fidelity technical implementation and engineering effort, we argue for low-fidelity, quick, and expressive explorations of a model's decision-making logic with model sketching. Just as work in end-user programming and programming by demonstration for AI systems has long worked towards enabling users to focus on *what* the system should do rather than *how* [9, 10], we seek to enable model authors to focus on *what* their models should reason over rather than dive too early into *how* their models are implemented when facing the task of problem formulation. Instead of waiting weeks or months until a model is implemented before evaluating in the real world and discovering that a problem formulation encodes harmful biases, we hope that ML practitioners can, in the course of a few days, instantiate multiple high-level model ideas and test them with users. Then, if they discover that their modeling approach is prone to bias in several ways, they can experiment with rounds of sketch models that might counter this bias, and only after this early-stage iteration would they move on to implement full production models. With model sketching, ML practitioners can move away from the mode of reactive implementation fixes and instead engage in the kind of proactive and creative model design exploration that can produce better models from the start.

ACKNOWLEDGMENTS

We thank our anonymous reviewers as well as Ranjay Krishna, Matthew Jörke, Mitchell Gordon, Lindsay Popowski, and Hancheng Cao for their valuable feedback on our paper. This work was partially supported by IBM as a founding member of the Stanford Institute for Human-centered Artificial Intelligence (HAI). Michelle S. Lam was supported by Stanford HAI and the Brown Institute for Media Innovation at the Stanford School of Engineering.

REFERENCES

[1] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the People: The Role of Humans in Interactive Machine Learning. *AI Magazine* 35, 4 (Dec. 2014), 105–120. https://doi.org/10.1609/aimag.v35i4.2513

[2] Apple. 2022. Create ML. https://developer.apple.com/machine-learning/create-ml/

[3] Phil Aquilina. 2022. Building Better Moderator Tools. https://www.reddit.com/r/RedditEng/comments/uly8s4/building_better_moderator_tools/

[4] Michael Brooks, Saleema Amershi, Bongshin Lee, Steven M. Drucker, Ashish Kapoor, and Patrice Simard. 2015. FeatureInsight: Visual support for error-driven feature ideation in text classification. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*. 105–112. https://doi.org/10.1109/VAST.2015.7347637

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[6] Bill Buxton. 2010. *Sketching User Experiences: Getting the Design Right and the Right Design.* Morgan Kaufmann.

[7] Joseph Chee Chang, Saleema Amershi, and Ece Kamar. 2017. Revolt: Collaborative Crowdsourcing for Labeling Machine Learning Datasets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 2334–2346. https://doi.org/10.1145/3025453.3026044

[8] Kathy Charmaz. 2006. *Constructing grounded theory: A practical guide through qualitative analysis.* Sage.

[9] Allen Cypher. 1991. EAGER: Programming Repetitive Tasks by Example. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New Orleans, Louisiana, USA) *(CHI '91)*. Association for Computing Machinery, New York, NY, USA, 33–39. https://doi.org/10.1145/108844.108850

[10] Allen Cypher, Daniel C. Halbert, David Kurlander, Henry Lieberman, David Maulsby, Brad A. Myers, and Alan Turransky (Eds.). 1993. *Watch What I Do: Programming by Demonstration.* MIT Press, Cambridge, MA, USA.

[11] Pedram Daee, Tomi Peltola, Aki Vehtari, and Samuel Kaski. 2018. User Modelling for Avoiding Overfitting in Interactive Knowledge Elicitation for Prediction. In *23rd International Conference on Intelligent User Interfaces* (Tokyo, Japan) *(IUI '18)*. Association for Computing Machinery, New York, NY, USA, 305–310. https://doi.org/10.1145/3172944.3172989

[12] Siddharth Dangi. 2020. Understanding dwell time to improve LinkedIn feed ranking. https://engineering.linkedin.com/blog/2020/understanding-feed-dwell-time

[13] Steven P. Dow, Alana Glassco, Jonathan Kass, Melissa Schwarz, Daniel L. Schwartz, and Scott R. Klemmer. 2011. Parallel Prototyping Leads to Better Design Results, More Divergence, and Increased Self-Efficacy. *ACM Trans. Comput.-Hum. Interact.* 17, 4, Article 18 (dec 2011), 24 pages. https://doi.org/10.1145/1879831.1879836

[14] Dean Eckles. 2022. Algorithmic transparency and assessing effects of algorithmic ranking. https://doi.org/10.31235/osf.io/c8za6

[15] Jerry Alan Fails and Dan R. Olsen. 2003. Interactive Machine Learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces* (Miami,

Florida, USA) *(IUI '03)*. Association for Computing Machinery, New York, NY, USA, 39–45. https://doi.org/10.1145/604045.604056

[16] Daniel Fallman. 2003. Design-Oriented Human-Computer Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) *(CHI '03)*. Association for Computing Machinery, New York, NY, USA, 225–232. https://doi.org/10.1145/642611.642652

[17] Rebecca Fiebrink, Dan Trueman, and Perry R. Cook. 2009. A Meta-Instrument for Interactive, On-the-fly Machine Learning. In *International Conference on New Interfaces for Musical Expression* (Pittsburgh, PA) *(NIME '09)*.

[18] James Fogarty, Desney Tan, Ashish Kapoor, and Simon Winder. 2008. CueFlik: Interactive Concept Learning in Image Search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 29–38. https://doi.org/10.1145/1357054.1357061

[19] Jules Françoise, Baptiste Caramiaux, and Téo Sanchez. 2021. Marcelle: Composing Interactive Machine Learning Workflows and Interfaces. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 39–53. https://doi.org/10.1145/3472749.3474734

[20] Vinod Goel. 1995. *Sketches of Thought*. MIT Press.

[21] Google. 2022. Teachable Machine. https://teachablemachine.withgoogle.com/

[22] Mitchell L. Gordon, Michelle S. Lam, Joon Sung Park, Kayur Patel, Jeff Hancock, Tatsunori Hashimoto, and Michael S. Bernstein. 2022. Jury Learning: Integrating Dissenting Voices into Machine Learning Models. 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 115, 19 pages. https://doi.org/10.1145/3491102.3502004

[23] Björn Hartmann, Scott R. Klemmer, Michael Bernstein, Leith Abdulla, Brandon Burr, Avi Robinson-Mosher, and Jennifer Gee. 2006. Reflective Physical Prototyping through Integrated Design, Test, and Analysis. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology* (Montreux, Switzerland) *(UIST '06)*. Association for Computing Machinery, New York, NY, USA, 299–308. https://doi.org/10.1145/1166253.1166300

[24] Fred Hohman, Kanit Wongsuphasawat, Mary Beth Kery, and Kayur Patel. 2020. Understanding and Visualizing Data Iteration in Machine Learning. 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, 1–13. https://doi.org/10.1145/3313831.3376177

[25] Youyang Hou and Dakuo Wang. 2017. Hacking with NPOs: Collaborative Analytics and Broker Roles in Civic Data Hackathons. *Proc. ACM Hum.-Comput. Interact.* 1, CSCW, Article 53 (dec 2017), 16 pages. https://doi.org/10.1145/3134688

[26] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. 2021. *OpenCLIP*. https://doi.org/10.5281/zenodo.5143773 If you use this software, please cite it as below..

[27] Abigail Z. Jacobs and Hanna Wallach. 2021. Measurement and Fairness. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (Virtual Event, Canada) *(FAccT '21)*. Association for Computing Machinery, New York, NY, USA, 375–385. https://doi.org/10.1145/3442188.3445901

[28] Ellen Jiang, Kristen Olson, Edwin Toh, Alejandra Molina, Aaron Donsbach, Michael Terry, and Carrie J Cai. 2022. PromptMaker: Prompt-Based Prototyping with Large Language Models. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, Article 35, 8 pages. https://doi.org/10.1145/3491101.3503564

[29] Eunice Jun, Melissa Birchfield, Nicole De Moura, Jeffrey Heer, and René Just. 2022. Hypothesis Formalization: Empirical Findings, Software Limitations, and Design Implications. *ACM Trans. Comput.-Hum. Interact.* 29, 1, Article 6 (Jan 2022), 28 pages. https://doi.org/10.1145/3476980

[30] Jongbin Jung, Connor Concannon, Ravi Shroff, Sharad Goel, and Daniel G. Goldstein. 2020. Simple rules to guide expert classifications. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 183, 3 (2020), 771–800. https://doi.org/10.1111/rssa.12576 arXiv:https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/rssa.12576

[31] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 2611–2624. https://proceedings.neurips.cc/paper/2020/file/1b84c4cee2b8b3d823b30e2d604b1878-Paper.pdf

[32] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory sayres. 2018. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 2668–2677. https://proceedings.mlr.press/v80/kim18d.html

[33] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. 2020. Concept Bottleneck Models. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 5338–5348. https://proceedings.mlr.press/v119/koh20a.html

[34] Todd Kulesza, Saleema Amershi, Rich Caruana, Danyel Fisher, and Denis Charles. 2014. Structured Labeling for Facilitating Concept Evolution in Machine Learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 3075–3084. https://doi.org/10.1145/2556288.2557238

[35] Deepak Kumar, Patrick Gage Kelley, Sunny Consolvo, Joshua Mason, Elie Bursztein, Zakir Durumeric, Kurt Thomas, and Michael Bailey. 2021. Designing Toxic Content Classification for a Diversity of Perspectives. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*. USENIX Association, 299–318. https://www.usenix.org/conference/soups2021/presentation/kumar

[36] Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. 2019. Measuring Bias in Contextualized Word Representations. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*. Association for Computational Linguistics, Florence, Italy, 166–172. https://doi.org/10.18653/v1/W19-3823

[37] Michelle S. Lam, Mitchell L. Gordon, Danaë Metaxa, Jeffrey T. Hancock, James A. Landay, and Michael S. Bernstein. 2022. End-User Audits: A System Empowering Communities to Lead Large-Scale Investigations of Harmful Algorithmic Behavior. *Proc. ACM Hum.-Comput. Interact.* 6, CSCW2, Article 512 (Nov 2022), 34 pages. https://doi.org/10.1145/3555625

[38] James A. Landay and Brad A. Myers. 1995. Interactive Sketching for the Early Stages of User Interface Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '95)*. ACM Press/Addison-Wesley Publishing Co., USA, 43–50. https://doi.org/10.1145/223904.223910

[39] Youn-Kyung Lim, Erik Stolterman, and Josh Tenenberg. 2008. The Anatomy of Prototypes: Prototypes as Filters, Prototypes as Manifestations of Design Ideas. *ACM Trans. Comput.-Hum. Interact.* 15, 2, Article 7 (jul 2008), 27 pages. https://doi.org/10.1145/1375761.1375762

[40] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, 142–150. https://aclanthology.org/P11-1015

[41] Yaoli Mao, Dakuo Wang, Michael Muller, Kush R. Varshney, Ioana Baldini, Casey Dugan, and Aleksandra Mojsilović. 2019. How Data Scientists Work Together With Domain Experts in Scientific Collaborations: To Find The Right Answer Or To Ask The Right Question? *Proc. ACM Hum.-Comput. Interact.* 3, GROUP, Article 237 (dec 2019), 23 pages. https://doi.org/10.1145/3361118

[42] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. 2019. Reliability and Inter-Rater Reliability in Qualitative Research: Norms and Guidelines for CSCW and HCI Practice. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 72 (nov 2019), 23 pages. https://doi.org/10.1145/3359174

[43] 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

[44] Felicia Ng, Jina Suh, and Gonzalo Ramos. 2020. Understanding and Supporting Knowledge Decomposition for Machine Teaching. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference* (Eindhoven, Netherlands) *(DIS '20)*. Association for Computing Machinery, New York, NY, USA, 1183–1194. https://doi.org/10.1145/3357236.3395454

[45] Samir Passi and Solon Barocas. 2019. Problem Formulation and Fairness. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (Atlanta, GA, USA) *(FAT* '19)*. Association for Computing Machinery, New York, NY, USA, 39–48. https://doi.org/10.1145/3287560.3287567

[46] Samir Passi and Steven Jackson. 2017. Data Vision: Learning to See Through Algorithmic Abstraction. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing* (Portland, Oregon, USA) *(CSCW '17)*. Association for Computing Machinery, New York, NY, USA, 2436–2447. https://doi.org/10.1145/2998181.2998331

[47] Samir Passi and Steven J. Jackson. 2018. Trust in Data Science: Collaboration, Translation, and Accountability in Corporate Data Science Projects. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 136 (nov 2018), 28 pages. https://doi.org/10.1145/3274405

[48] Kayur Patel, Naomi Bancroft, Steven M. Drucker, James Fogarty, Amy J. Ko, and James Landay. 2010. Gestalt: Integrated Support for Implementation and Analysis in Machine Learning. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA) *(UIST '10)*. Association for Computing Machinery, New York, NY, USA, 37–46. https://doi.org/10.1145/1866029.1866038

[49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
[50] David Piorkowski, Soya Park, April Yi Wang, Dakuo Wang, Michael Muller, and Felix Portnoy. 2021. How AI Developers Overcome Communication Challenges in a Multidisciplinary Team: A Case Study. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW1, Article 131 (apr 2021), 25 pages. https://doi.org/10.1145/3449205
[51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. https://doi.org/10.48550/ARXIV.2103.00020
[52] Gonzalo Ramos, Christopher Meek, Patrice Simard, Jina Suh, and Soroush Ghorashi. 2020. Interactive machine teaching: a human-centered approach to building machine-learned models. *Human–Computer Interaction* 35, 5-6 (2020), 413–451.
[53] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10684–10695.
[54] 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
[55] Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M Aroyo. 2021. "Everyone Wants to Do the Model Work, Not the Data Work": Data Cascades in High-Stakes AI. 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 39, 15 pages. https://doi.org/10.1145/3411764.3445518
[56] Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. The Woman Worked as a Babysitter: On Biases in Language Generation. In *EMNLP/IJCNLP (1)*. 3405–3410. https://doi.org/10.18653/v1/D19-1339
[57] Patrice Simard, Saleema Amershi, Max Chickering, Alicia Edelman Pelton, Soroush Ghorashi, Chris Meek, Gonzalo Ramos, Jina Suh, Johan Verwey, Mo Wang, and John Wernsing. 2017. *Machine Teaching: A New Paradigm for Building Machine Learning Systems.* Technical Report MSR-TR-2017-26. https://www.microsoft.com/en-us/research/publication/machine-teaching-new-paradigm-building-machine-learning-systems/
[58] Masaki Suwa and Barbara Tversky. 1996. What Architects See in Their Sketches: Implications for Design Tools. In *Conference Companion on Human Factors in Computing Systems* (Vancouver, British Columbia, Canada) *(CHI '96)*. Association for Computing Machinery, New York, NY, USA, 191–192. https://doi.org/10.1145/257089.257255
[59] Maryam Tohidi, William Buxton, Ronald Baecker, and Abigail Sellen. 2006. Getting the Right Design and the Design Right. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada) *(CHI '06)*. Association for Computing Machinery, New York, NY, USA, 1243–1252. https://doi.org/10.1145/1124772.1124960
[60] Leandro von Werra. 2021. distilbert-imdb. https://huggingface.co/lvwerra/distilbert-imdb
[61] Miriam Walker, Leila Takayama, and James A. Landay. 2002. High-Fidelity or Low-Fidelity, Paper or Computer? Choosing Attributes when Testing Web Prototypes. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 46, 5 (2002), 661–665. https://doi.org/10.1177/154193120204600513 arXiv:https://doi.org/10.1177/154193120204600513
[62] Dakuo Wang, Justin D. Weisz, Michael Muller, Parikshit Ram, Werner Geyer, Casey Dugan, Yla Tausczik, Horst Samulowitz, and Alexander Gray. 2019. Human-AI Collaboration in Data Science: Exploring Data Scientists' Perceptions of Automated AI. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 211 (nov 2019), 24 pages. https://doi.org/10.1145/3359313
[63] Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts. 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 385, 22 pages. https://doi.org/10.1145/3491102.3517582
[64] Tongshuang Wu, Daniel S. Weld, and Jeffrey Heer. 2019. Local Decision Pitfalls in Interactive Machine Learning: An Investigation into Feature Selection in Sentiment Analysis. *ACM Trans. Comput.-Hum. Interact.* 26, 4, Article 24 (jun 2019), 27 pages. https://doi.org/10.1145/3319616
[65] Maria C. Yang. 2005. A study of prototypes, design activity, and design outcome. *Design Studies* 26, 6 (2005), 649–669. https://doi.org/10.1016/j.destud.2005.04.005
[66] Qian Yang, Justin Cranshaw, Saleema Amershi, Shamsi T. Iqbal, and Jaime Teevan. 2019. Sketching NLP: A Case Study of Exploring the Right Things To Design with Language Intelligence. 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/10.1145/3290605.3300415
[67] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. 2020. Re-Examining Whether, Why, and How Human-AI Interaction Is Uniquely Difficult to Design. 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, 1–13. https://doi.org/10.1145/3313831.3376301
[68] Amy X. Zhang, Michael Muller, and Dakuo Wang. 2020. How Do Data Science Workers Collaborate? Roles, Workflows, and Tools. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW1, Article 22 (may 2020), 23 pages. https://doi.org/10.1145/3392826
[69] Ron Zhu. 2020. Enhance Multimodal Transformer With External Label And In-Domain Pretrain: Hateful Meme Challenge Winning Solution. https://doi.org/10.48550/ARXIV.2012.08290

A MODELSKETCHBOOK PROCESS OVERVIEW

Here, we summarize the three commands that a user executes to set up their first sketch model in a computational notebook. First, the user sets up their sketchbook, as previously shown in Section 3.2.1. They only need to perform this operation once to load their data, and they can create as many concepts and sketches as they would like using this single sketchbook.

```
import model_sketch_book as msb

# Set up the sketchbook
sb = msb.create_model_sketchbook(
    goal="Detect hateful memes on social media.",
    datasets={  # Pass in datasets to use
        "train": df_train,
        "test": df_test,
    },
    schema = {  # Specify the types of data fields
        "text": InputType.Text,
        "img_url": InputType.Image,
        "overall_rating": InputType.GroundTruth,
    },
)
```

Next, they can create as many concepts as they would like using calls to the `create_concept_model()` function as shown below. The function parameters can be equivalently specified using UI widgets, and the function returns a cell output with a dataframe visualization of the concept results (Figure 5).

```
msb.create_concept_model(
    input_field="post_image",  # Column for input
    concept_term="cartoon",    # Concept description
)
```

Finally, the user can create as many sketches as they would like by making calls to `create_sketch_model()` as shown below; again, the user can alternatively specify the same function parameters using widgets, and the function displays a cell output with a dataframe visualization of the sketch model results (Figure 6).

```
msb.create_sketch_model(
    concepts=["cartoon", "rant"],  # To aggregate
    sketch_id="sketch_version_1",  # Sketch name
)
```

B MODEL SKETCHING CASE STUDIES

Table 5 summarizes the task and dataset for each example model sketching application and includes concepts and sample sketches.

	Travel planning	Creative inspiration	Political candidate research	Reviewer bias audit
Task	Assist travelers with selecting an Airbnb for a trip to New York City.	Help users to uncover creative connections across artists and artworks.	Assist voters with finding a candidate for the 2022 California gubernatorial election who aligns with them on issues that matter most.	Assist a food reviewer in checking their own biases in reviewing cafes and restaurants on social media.
Data	Airbnb listings in NYC. Includes the name, image, description, and neighborhood overview for each listing.	Museum artwork pieces. Includes the artist's name, title, type, style, genre, media, geographic region, image and text description for each artwork.	Candidate information aggregated from political statements and speeches by CalMatters, a nonprofit, nonpartisan, third-party group.	Instagram food reviews. Includes name of the cafe/restaurant, post date, post text, number of likes, number of comments, and overall rating.
Concepts	Trendy, Central, Clean, Dingy, Tidy, Designed well, Designer, Bright, Outdated, Sunlight natural, Spacious, Ugly, Things to do, Activities, Good vibes	Colorful, Surreal, Fantasy, Abstract, Realist, Politics themed, Revolution themed, Labor movement themed, Social justice themed, Race themed, Gender themed, Feminist, Queer themed	Environment, Equality, Gun control, Reproductive rights, Voting rights	Tasty, Affordable, Creative, Nice vibes, Good service, High quality, Good presentation, Yummy, Cheap, Original, Decent vibes, Sound service
Sample Sketches	*Aesthetic*: {Clean, Bright, Designer}	*Bold*: {Colorful, Surreal, Fantasy, Abstract}	*High-level issues*: {Climate change, Gun control}	*Naive concepts*: {Tasty, High quality, Affordable, Creative, Nice vibes, Good service, Good presentation}
	Fun neighborhood: {Good vibes, Central, Trendy}	*Social justice*: {Social justice themed, Race themed, Feminist, Queer themed}	*Issues & opinion*: {Eco-friendly, Anti gun, Equal rights}	*Refined concepts*: {Yummy, High quality, Cheap, Original}
	Modern beauty: {Outdated, Ugly, Clean}	*Political movements*: {Politics themed, Revolution themed, Labor movement themed}	*Issues & detailed opinion*: {Eco-friendly, Anti gun, Pro choice, Equality, Voting Rights}	*Refined concepts, extended*: {Yummy, High quality, Cheap, Original, Decent vibes, Sound service, Good presentation}

Table 5: A summary of several model sketching applications. We explored a task that assisted users in selecting an Airbnb for travel planning, a task aimed at surfacing creative connections between artists and artworks, a task designed to assist voters in discovering their alignment with political candidates, and a task that allowed a food reviewer to investigate the factors they value most and potential biases they might hold.

C QUALITATIVE ANALYSIS

For the free-text written responses, we sought to summarize the high-level themes that emerged from our participants, so the codes were not the product, but our process [42]. The first author conducted an inductive analysis to summarize participants' model authoring experiences. The first step of this process was to read through all written responses multiple times. Then, the qualitative open coding [8] process was iterative and took place in two phases: the first phase involved line-by-line response coding to closely reflect the original data (e.g., "noticed minimal representation of violence in the dataset," "plans to use a transformer model," or "there was inconsistency among the labels"). Then, the second phase synthesized codes from the first phase into higher level themes (e.g., "data representativity" or "labeling methodology"). After this theme generation, written responses were coded based on the themes to characterize participants' planned modeling approaches and their learnings and takeaways from the task.

D PRE- AND POST-TASK WRITTEN RESPONSE QUESTIONS

D.1 Pre-task

Participants were asked to provide written responses for the following questions prior to the main model sketching task.

(1) (Before tutorial) **Describe your modeling approach.** Please write a few sentences describing your ideas on how you might build a model to detect posts that should be removed from a social media platform.
   - Approach:
   - Time estimate (how long do you think it would take to complete this planned approach?):

(2) (After tutorial) **Initial concept brainstorming.** Please list 3 concepts that you think may be relevant to your task.

D.2 Post-task

Participants were asked to provide written responses for the following questions immediately after the main task.

(1) **Describe your modeling approach (again).** After going through today's session, please write a few sentences describing how you would plan to build a model that detects posts that should be removed from a social media platform. You may assume you have access to a tool like ModelSketchBook, but you are not required to use such a tool.
(2) **Learnings.** Based on your explorations today, is there anything new that you've learned about the task, the data, or your modeling goals? Are there any gaps you noticed with the data or the task formulation? Are there any changes you'd like to make to address these gaps?
(3) **Choose your favorite sketch.** Please run the cell directly below this to see all of the sketches you authored. Then, in the cell below that, please enter the sketch ID of the sketch that you feel performed the best (or most closely matched

your modeling goals). Finally, run the last cell of the notebook.

E POST-TASK SURVEY QUESTIONS

The following are the questions that participants were asked to respond to in a survey after the main task.

(1) How **satisfied** are you with the **concept models** that you created using the API? (single-select options: Very dissatisfied, Dissatisfied, Somewhat dissatisfied, Neither dissatisfied nor satisfied, Somewhat satisfied, Satisfied, Very satisfied)
(2) How **satisfied** are you with the **sketch models** that you created using the API? (single-select options: Very dissatisfied, Dissatisfied, Somewhat dissatisfied, Neither dissatisfied nor satisfied, Somewhat satisfied, Satisfied, Very satisfied)
(3) To what extent do you feel that your approach to the modeling task **changed** as you experimented with and iterated upon your concepts and sketches? (single-select options: No changes, A few changes, Some changes, Many changes)
(4) To what extent do you feel that your approach to the modeling task **improved** as you experimented with and iterated upon your concepts and sketches? (No improvement, Slight improvement, Noticeable improvement, Significant improvement)
(5) How would you rate your **comfort** in using tools like Scikit-learn, PyTorch, or TensorFlow to author ML models? (Very uncomfortable, Uncomfortable, Somewhat uncomfortable, Neither uncomfortable nor comfortable, Somewhat comfortable, Comfortable, Very comfortable)
(6) How would you rate your **comfort** in using the ModelSketchBook API? (Very uncomfortable, Uncomfortable, Somewhat uncomfortable, Neither uncomfortable nor comfortable, Somewhat comfortable, Comfortable, Very comfortable)
(7) How would you rate your ability to express your voice/intent using tools like Scikit-learn, PyTorch, or TensorFlow to author ML models? (Very inexpressive, Inexpressive, Somewhat inexpressive, Neither inexpressive nor expressive, Somewhat expressive, Expressive, Very expressive)
(8) How would you rate your ability to express your voice/intent when using the ModelSketchBook API? (Very inexpressive, Inexpressive, Somewhat inexpressive, Neither inexpressive nor expressive, Somewhat expressive, Expressive, Very expressive)
(9) What did you **like** about your model authoring experience with the ModelSketchBook API? Please describe the positives of this API.
(10) What did you **not like** about your model authoring experience with the ModelSketchBook API? Please describe the negatives of this API or any issues you experienced.
(11) (Optional) Feel free to write any other comments, feedback, or thoughts you may have.

F POST-TASK INTERVIEW QUESTIONS

The following are the questions that we asked participants in an interview after the main task. We also asked follow-up questions as necessary.

(1) **Thought process.** Could you walk me through your notebook and explain your thought process as you designed your concepts and sketch models?
(2) **Contrast.** How did this model authoring process differ from your prior experience authoring other models?
   - What did you like most about it?
   - What did you like least about it?
(3) **Learnings.** What did you learn from this model sketching process? Do you have any takeaways or perspective changes?

G PARTICIPANT DEMOGRAPHICS

We had 3 women, 12 men, 1 non-binary participant, and 1 participant who preferred not to share their gender; we had 14 Asian participants, 2 White participants, and 1 participant who identified as White and Hispanic. We had 9 participants aged 18-24 and 8 aged 25-34. For highest educational attainment, 9 participants had earned a doctorate, master's or other professional degree, 4 had earned a four-year degree, 3 had completed some college education, and 1 had graduated from high school. There were 9 participants who identified as ML engineers or data scientists working in industry, and there were 8 participants who identified as students who use ML in their work or research. Of the industry practitioners, 3 were employees of startup companies, and 6 were employees of large companies (with 1000+ employees). Participants reported an average of 3.6 years of experience working with machine learning.

H MODELING GAPS

Table 6 summarizes the main categories of modeling gaps that participants identified along with direct participant quotes.

I PARTICIPANTS' THOUGHT PROCESS

Participants perceived a change in their mode of thinking when they engaged in model sketching. Many participants felt that the approach freed them from the tedious tasks that they typically had to work on. P10 described model sketching as "a different way of thinking about [modeling]" compared to their day-to-day modeling work, which was "a lot more manual and technical and a lot more work and a lot more complex," so they liked how model sketching "just does it for you." Similarly, P5 expressed that model sketching "outsourced a lot of the most annoying parts of ML, like creating pipelines to generate features."

Some participants felt that this approach allowed them to take a more *long-range view* of the modeling process: P1 stated that "this process helps put into perspective the long term goals and the structure of the ML lifecycle." Others described the model sketching process as refreshingly *creative* and *artistic*, while their typical model authoring experiences had felt more formulaic. As P12 put it, "[model sketching] forced me to think creatively in a way I don't normally have to. Normally, my work is pretty boilerplate-y and repetitive. In this case, the model's performance was much more limited by my own creativity and ability, so it felt more like I had artistic direction in how the model did."

Participants also enjoyed the interpretability of model sketching, which afforded them a greater understanding of model behavior compared to past model authoring experiences. P14 shared that

Category	Learning or Takeaway	Quotes
Task	Text overall seems more important than images for this task	"The image could be detached from the text and the text seemed a bit more useful in this exercise." (P7)
	Depending on the case, it may be important to combine the modalities or focus on text or image	"How text and images interact together is a super important takeaway. You can't just think about the images alone; you can't just think about the text alone, it's together the effect that they create." (P16), "One counter to [the greater importance of text] might be images of gore, where text is not necessarily needed for the post to be problematic." (P2)
	Memes are tricky to understand and label	"Memes are tricky in that the images may be benign but the text may be harmful." (P1), "Memes are very vague and tricky to label. The subtext is hard to capture with simple concepts." (P6)
Dataset	There is a need for greater representativity of kinds of harm in the data (homophobia, transphobia, nudity)	"Need more data [to capture] problematic posts of all kinds (not just racism or sexism) [...] Not enough data for homophobia, transphobia etc." (P1), "[...] in this case nudity wasn't an issue, but might be for test time." (P16)
	The dataset is may be too unrealistic and contrived to match the online distribution of posts	"The data seems to be far off from what is normally posted online, so the data definitely seems not good for general use." (P8)
Labels	Class imbalance has a large impact on modeling and makes it difficult to calibrate a model	"I learned that class imbalance plays a large impact on modeling—it seemed for example that my own labeled dataset was relatively weighted towards not removing posts, so it paid dividends to [try] more specific targeted terms towards things that I found truly offensive like advocating for violence." (P5)
	Their own labels appeared to be internally inconsistent	"My labels are obviously not very internally consistent either and collecting from larger samples may help to even out the noise." (P7)
	It would be valuable to increase the removal-score granularity beyond binarization	"One area of improvement could be to increase the task fidelity to actually use the 5 labels instead of binarizing them." (P3)

Table 6: After performing a model sketching task, participants noted takeaways for their modeling approach and identified gaps in the dataset and labeling methodology.

"usually a lot of what I do is just a black box [...] and it's frustrating when it doesn't work," but that "having this other option of trying to build from the bottom up is a pretty good alternative." This modeling approach also allowed participants to think about decision-making from first principles. P16 described that "in terms of reasoning at a human level, I really like this idea of breaking it down into smaller concepts. It seems like an intuitive way you'd teach a kid... to me that just seems to be building a model that's more explainable."

J PARTICIPANT EXPERIENCE OUTCOMES

Addressing RQ2 in terms of benefits to ML practitioners, we observed that participants found the model sketching approach to be comfortable and expressive. Participants shared a great deal of excitement about this method of model development, expressing that they "enjoyed the experience of thinking about models in this manner" (P1) and that the tool "could be really helpful in the future for similar tasks" (P11) or for "showing that something was possible and expanding imaginations" (P12).

J.1 Comparison with existing authoring tools

Comparing to their experiences with more common model authoring tools like Scikit-learn, PyTorch, and TensorFlow, participants felt similar levels of comfort with the ModelSketchBook API (Figure 12) even though this 30-minute task was their first exposure to our tool, and all participants had significant experience with model authoring using these more traditional tools. At the same time, we saw that participants reported higher levels of expressivity when using the ModelSketchBook API than when using those same common model authoring tools. These results present promising support that our model sketching approach can be successfully adopted by ML practitioners as a lightweight, expressive entrypoint to model authoring.

J.2 Reflections on the model sketching experience

In our survey, we also probed users' levels of satisfaction with their concepts and sketch models and asked them about the extent to which they felt their modeling approach changed or improved over time. We observed that participants were highly satisfied with both their concepts and sketch models. All but two participants rated their satisfaction with both models as "somewhat satisfied," "satisfied," or "very satisfied." All but one participant responded that their modeling approach changed as they iterated on concepts and sketches, and the vast majority of participants (14 of 17) described a "slight improvement," "noticeable improvement," or "significant improvement" in their modeling approach.

In their interview feedback and survey free-responses, participants expressed enthusiasm about the model sketching approach (Table 7). The most frequently-cited benefits of model sketching were that it was highly interpretable and explainable, easy to use (especially because it required no code), provided fast feedback loops, and prompted them to take on a different, high-level mode of thought. Meanwhile, the main limitations that participants noted were that the notebook environment often became disorganized, that our example-centric (rather than metric-centric) visualizations required more time to interpret, and that they would appreciate more assistance on brainstorming concepts. Multiple participants were excited about the opportunity to allow non-technical users and domain experts who lack ML knowledge to use ModelSketchBook and leverage their domain-specific expertise. They also suggested

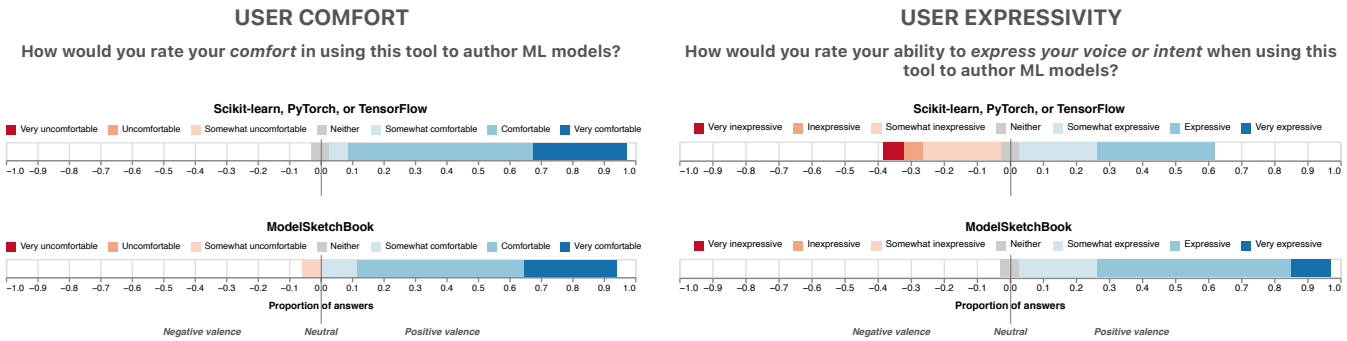

Figure 12: Upon their first-time use of the ModelSketchBook API, participants indicated similar levels of comfort as they feel with traditional model authoring tools (left), and they indicated higher levels of expressivity using ModelSketchBook (right).

that we go beyond our notebook prototype to set up a dedicated user interface, and they suggested helper tools that would address potential issues with bias and overfitting in concepts.

K SOTA MODEL EVALUATION

K.1 Benchmark Dataset and Task Details

For the movie review sentiment task, we gathered our data from the IMDB Movie Reviews dataset [40], a binary sentiment analysis dataset. Each of the examples in this dataset is a review drawn from the Internet Movie Database (IMDB) that has been labeled as positive or negative sentiment. This dataset is balanced between positive and negative labels, and it was curated to only contain polarizing reviews with negative reviews of 4 or below and positive reviews of 7 or above on a 10-point rating scale. The SOTA model against which we compare our sketch models is `distilbert-imdb` [60], which reported 0.928 accuracy on this binary classification task.

For the comment toxicity task, we used the dataset from Kumar et al. [35] that gathered diverse perspectives on the toxicity of comments drawn from Reddit, Twitter, and 4chan. The dataset features ratings from 17,280 U.S. survey participants. Each item consists of a comment and a particular participant's toxicity rating for that comment, using a 5-point rating scale from "Not at all toxic" to "Extremely toxic." This dataset consists of over 100,000 examples and is skewed toward lower-toxicity ratings. A key focus of the Kumar et al. work was that there are diverse perspectives on what constitutes toxic content, which drives substantial disagreement among raters in the dataset and results in a challenging toxicity prediction task. We formulated this task to predict the aggregate rating for each item across annotators since the sketch model was designed to predict at the item level rather than the (item, annotator) level. Thus, the SOTA model for this task is the aggregated model variant from Gordon et al. [22], which reported 0.90 MAE.

Finally, for the Hateful Memes task, we drew from the full training set from Kiela et al. [31], described in Section 5.1.2. In contrast to the dataset sample we used in the study, here we did not filter to the examples with "non-hateful" labels, and we used the centralized "hateful meme" definition provided by Meta AI rather than building on participants' own preferred definitions. The SOTA model for this task was drawn from first place solution in the Hateful Memes Challenge [69], which reported 0.845 AUROC, and we also compare against the 0.714 AUROC SOTA reported by Meta AI at the time of the dataset release.

K.2 Sketch Model Details

For each of the task datasets, we randomly sampled up to 500 examples to train our sketch models. In comparison, the SOTA models for these tasks were trained on 25,000 examples for the IMDB task, 100,000 for the comment toxicity, and 8,500 examples for the Hateful Memes task. Then, for each task, a member of our team iteratively created sketch models based solely on the sampled training set, attempting to create the most performant sketch models possible based on the training data.

For the movie review sentiment task, this process resulted in a final sketch model with 16 concepts: enjoyable, boring, mediocre, engaging, overrated, underwhelming, award winning, time waste, hate, love, positive sentiment, negative sentiment, sarcasm, praise, compliment, and mocking. Since this dataset consists only of movie review text, all of the concepts were text concepts. This sketch model used linear regression to aggregate the specified concepts. Then, for the comment toxicity task, there were 14 concepts in the final sketch model: racism, sexism, homophobic, transphobic, ableist, religious discrimination, toxicity, hurtful, threat, misinformation, trolling, xenophobia, kind, and harmless. Again, since all examples in this dataset were text comments, all of the concepts were text concepts. This sketch model also used linear regression to aggregate the concepts. Lastly, for the Hateful Memes task, there were 7 concepts in the final sketch model: racist, hate speech, vulgar, harmless, religious, sexist, and violence. All of these were text concepts based on the meme text; in line with our study participants, we found that image concepts did not work as well as text concepts for this task. This sketch model used a multi-layer perceptron (MLP) classifier with two hidden layers to aggregate the selected concepts.

Finally, to evaluate sketch model performance, we used a random sample of 100 examples from the test set (preserving the same label distribution as the full test set). We applied the trained sketch models to this test set to generate sketch model predictions. We then compared these predictions against the ground truth labels to calculate the performance metric matching the headline performance metric reported by each of the tasks' SOTA models.

Category	Theme	Quotes
Likes	Easy to use and get started	"...the thing I liked the most was how quickly I was able to get started, like I ran a few lines of code and immediately I was getting some results. So I think ModelSketchBook really makes machine learning more accessible which is really important." (P9), "I like that I didn't have to write any code for the task." (P5), "It's easy to get started and less prone to error." (P7), "This provides a really solid baseline without writing a single line of code." (P3)
	Interpretable, explainable	"Very interactive and also very interpretable which is something with ML models I usually work with is not the case. So this is super super helpful. It's like a different way of thinking about it." (P10), "In terms of interpretability of the decision and explainability, which is also important in this kind of task, it [ModelSketchBook] is very helpful." (P4), "I like the semantic idea of a concept because it really puts model interpretability at the forefront and I think from a lot of experience, sometimes you try a bunch of models and you're not really thinking about them as semantic concepts from the get-go." (P16)
	Fast feedback loops	"This modeling process had a lot more quick feedback loops and iteration of feature exploration and then turning that into models pretty fast, which is a good thing for sure." (P5), "In my previous models at work, we do ad targeting, so it's very hard to get this kind of real-time feedback." (P6)
	Different, high-level way of thinking	"I liked the idea to approach things from how a human will think about things by defining these concepts." (P16), "...focused around ideas and thinking about the task rather than hyperparameters or technical ML details." (P10), "This process helps put into perspective the long term goals and the structure of the ML lifecycle and the pipeline that you need to create: what kind of data you need, which we prioritize, where textual data can provide more insight than image data and vice versa." (P1)
Dislikes	Notebook interface felt messy at times	"a bit annoying to copy paste some of the cells." (P5)
	Takes time to sift through examples to gauge performance	"hard to go through each example because it's long." (P6), "it would be nice to not have all the examples pop up every time just to gauge performance." (P16)
	Needed more guidance to brainstorm concepts	"it is hard to make accurate concepts [...] that is more of an issue with choosing relevant features by hand." (P9), "Not much experience/guidance on how to come up with concepts creatively." (P10), "I wasn't sure how to improve the model." (P12)
Suggestions	Building a dedicated UI	"I think taking this out of IPython and having a more native UI would be better." (P3), "I think having a GUI interface would be nice." (P16)
	A version tailored for non-technical users and domain experts	"I think the API is great for model developers who do not have a formal background with ML or deep learning. This API is a super useful tool for folks for product engineers and product managers." (P3). "I feel like where this could be really relevant and useful is for folks that don't know much about ML, but know a lot about the task at hand, and for them to get something working in their hands." (P12)
	Providing tools to address overfitting and bias	"It's easy to fall into the trap of just trying to optimize your performance on the examples you're seeing and then ... forget what the rest of the data looks like." (P8), "Human intervention in designing concepts can be biased." (P4), "Perhaps I would miss edge cases." (P12)
	Allowing for more manual control of weights and model types	"I think a lot of the sketches could be improved by adding more weights to a concept." (P13), "Lacking in features a bit (nonlinear models, more types of signal)." (P12)

Table 7: In written and interview responses, participants shared that they liked the process and cognitive mode that model sketching unlocked. They felt that the tool could be improved with a dedicated UI and scaffolding to assist concept brainstorming.