# Local Decision Pitfalls in Interactive Machine Learning: An Investigation into Feature Selection in Sentiment Analysis

TONGSHUANG WU, DANIEL S. WELD, and JEFFREY HEER, University of Washington

Tools for Interactive Machine Learning (IML) enable end users to update models in a "rapid, focused, and incremental"—yet local—manner. In this work, we study the question of local decision making in an IML context around feature selection for a sentiment classification task. Specifically, we characterize the utility of interactive feature selection through a combination of human-subjects experiments and computational simulations. We find that, in expectation, interactive modification fails to improve model performance and may hamper generalization due to overfitting. We examine how these trends are affected by the dataset, learning algorithm, and the training set size. Across these factors we observe consistent generalization issues. Our results suggest that rapid iterations with IML systems can be dangerous if they encourage local actions divorced from global context, degrading overall model performance. We conclude by discussing the implications of our feature selection results to the broader area of IML systems and research.

CCS Concepts: • **Human-centered computing** → **Empirical studies in HCI**; Visualization design and evaluation methods; • **Computing methodologies** → *Feature selection*;

Additional Key Words and Phrases: Machine learning, text classification, performance analysis

**ACM Reference format:**
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 (June 2019), 27 pages.
https://doi.org/10.1145/3319616



## 1 INTRODUCTION

Interactive machine learning (IML) systems aim to ease the process of training a model by providing tools that support more rapid, focused, and incremental model updates than seen in a traditional machine learning (ML) process [2]. These properties enable everyday users to interactively explore the model space through trial-and-error and drive the system toward an intended behavior, hopefully reducing the need for supervision by ML experts [14, 30, 63]. IML enables two-way interaction between human and machines: On one hand, the system explains to users how the learner is making predictions, usually through visual [38, 41] or textual [34] feedback on model performance [10, 52, 63]. On the other hand, the user then communicates modifications back to

The project was supported by the Moore Foundation Data-Driven Discovery Investigator program, with additional support from ONR grant N00014-18-1-2193, the WRF/Cable Professorship and Google.
Authors' addresses: T. Wu, D. S. Weld, and J. Heer, University of Washington, 185 E Stevens Way NE, Seattle, WA 98195; emails: {wtshuang, weld, jheer}@cs.washington.edu.

1073-0516/2019/06-ART24 $15.00
https://doi.org/10.1145/3319616

the learning system to enhance the resulting model. For example, the user may label more data points [25, 57], adjust learning parameters [27], or add and delete features [33, 63].

While IML may help analysts express their domain knowledge [60], and labeling additional data is likely to improve performance, we suspect that certain IML approaches may actually hurt the resulting models due to the human desire for local improvements. Indeed, prior work [56] documents the human tendency to anchor on available contextual information and to focus heavily on local changes. In fact, local decisions, the ones we make without having access to or considering all the information available, are ubiquitous. As claimed in bounded rationality [58], individuals' decisions are naturally limited by the tractability of the decision problem, the cognitive limitations of their minds, and the time available to make the decision. Many decision biases reflect the narrow locality of the decision making context. For example, Tversky and Kahneman [65] describe how we tend to rely on one specific piece of information, while O'Donoghue and Rabin [44] note that we over-value immediate rewards. It has also been reported that these decisions, while compelling locally, can add up to inferior overall patterns [56].

The prevalence of local decisions in the wild makes us suspect that locality is also a potential denominator in IML. As a highly iterative and exploratory process, the rapid updates enabled by IML tools may exacerbate practitioners' focus on local information and create the danger of inadvertent overfitting. For example, a practitioner may attempt to fix specific errors presented by a tool, overlooking the global effect of their actions.

More specifically, IML users may fall prey to either *over-confidence* or *over-reliance* [59]. In the first case, users may trust in their own domain knowledge too strongly, lowering learner performance by acting on their intuitions [37, 60]. In the second case, they may overly rely on IML system feedback specific to the current model and overfit to the training or validation dataset [10, 63]. Both over-confidence and over-reliance may hamper learner generalization due to localized decisions. While some prior work has observed decreased model performance resulting from human input [2, 60], the role of IML in such problems has yet to be formally studied: To what degree does IML help improve models? Is decreased model performance a small probability event, or a fairly common consequence that researchers should be aware of?

This article seeks to illuminate potential IML pitfalls that may arise when users are encouraged to immediately react to local observations about their model. Evaluation on IML is challenging, as it can involve diverse domains and algorithms. A natural first step, therefore, is a closer examination in the context of a concrete domain. Specifically, we base our study on interactive feature selection, which takes the form of adding or deleting certain features, within a text sentiment classification task. Text data is a common modality in IML research [10, 30, 33, 41, 55, 60, 61, 63], and sentiment analysis is an interesting and representative task. Feature selection, while just one activity supported by IML systems, is provided by many IML systems [10, 15, 33, 55, 60, 61, 63]. It also focuses on the immediate current iteration of a learner: classic feature selection systems in the wild typically show the performance of the most up-to-date model that users have created with previous manipulations. User input is by definition local when users make decisions based on iteration-sensitive feedback.

We first investigate how people make local feature selection decisions, and assess the impacts of these decisions on model performance. Specifically, we examine people's judgments on whether to include or exclude features (text phrases) in a sentiment analysis context. We focus on how subjects balance prior knowledge (e.g., word polarity) and feedback on model performance (e.g., change in F1 classification score) in their decision process. Through human-subjects experiments with both crowdworkers and ML practitioners, we find that both factors sway users' decisions, but prior knowledge proves more influential. We also observe that participants become less confident and spend more time on cases where the model does not perform as they expect. Importantly,

interactive input by ML practitioners *fails to improve models on average.* The user impact on models seems to be largely random. Even for individuals who successfully reach an improved model, their model performance oscillates as they revise the feature space.

Informed by our experiment, we next define several automated strategies for interactive feature selection, and then use them to simulate user input and study their effects at scale. We manipulate factors including decision criteria, dataset, training set size, learning algorithm, and (where applicable) regularization parameter. Across all feature selection strategies we see no significant improvement in model performance on held-out test data. In a few isolated cases (e.g., extreme regularization) we observe that user input may provide a helpful boost. However, in those cases, the final model performance is poor (i.e., a user can make a bad model slightly better) and not competitive with alternative models fit without human intervention.

Our results suggest that classical classification algorithms do not benefit from interactive local modification of low-level features. Despite the constrained setting, we see our work as a baseline for understanding the impact of IML in which both prior knowledge and immediate model performance feedbacks are taken into consideration. Assuming users are guided by stage-sensitive feedback that prompts iterative local updates, we predict that human intervention is likely to be inefficient, and even potentially harmful.

To be clear, we are *not* claiming that IML "fails" or "has little use." Our study is limited to interactive feature selection for text sentiment classification. While our results raise a cautionary note, they may not generalize to other activities or domains. To fully build a "checklist" for when, where, and how IML systems should be used, additional work is needed on how various factors affect the utility of IML in different tasks. We conclude the article with suggestions for future research in IML that may shift focus away from overly "localized" perspectives and interventions. We hypothesize that IML systems may be improved by (1) providing global feedback to present a more comprehensive picture of the models before inviting user intervention, and (2) engaging users in higher level interactions that leverage domain expertise rather than low-level "tuning."

In summary, our research contributions are as follows:

- The results of a controlled human-subjects experiment on interactive feature selection. We identify user strategies for balancing prior knowledge with model performance feedback and find that interactive feature selection fails to improve model generalization.
- The results of a large-scale simulation of user feature selection decisions across datasets, training set sizes, learning algorithms, and hyper-parameters, informed by our first study. We confirm that across varied contexts and algorithms, interactive feature selection fails to improve models on average.
- A discussion of the implications of our results for IML systems and research. We suggest that funneling user attention toward iterative, local model refinement is unlikely to yield improved models. Rather, we argue for focusing user attention toward tasks that can reliably convey high-level domain expertise and ethical oversight, including training data quality assessment and model error analysis.

## 2 RELATED WORK

We first provide an overview of IML. We then focus on aspects of IML that inspire our study: feature selection, user decision factors, and performance impacts.

### 2.1 State-of-the-Art IML Systems

IML is a field that actively includes humans in ML model construction procedures. Broadly speaking, most IML systems seek to either (1) enable non-programmers to build ML systems [4, 18, 19],

(2) help end users understand their model failures [1, 3], and (3) improve their models [25, 30, 61]. As the name indicates, it enables two-way interaction between human and machines: the system explains to users how it makes predictions, and the user then communicates modifications back to the learning system [33].

Researchers have explored both sides. On one hand, many IML systems seek to expressively present model behaviors. For instance, ModelTracker [3] and Squares [52] map the prediction confidence of classifiers to locate false positives/negatives and quantify the level of "incorrectness" for users. Similarly, Alsallakh et al. [1] analyze probabilistic distributions in classifiers to understand relations of different classes. Besides visual representations, prior work also inspects model behaviors with "what-if" scenarios. Users are allowed to either input new instances [37] or manipulate certain features for a given instance [27, 31, 47].

On the other hand, researchers have also tried to understand what kind of human feedback is most effective. Labeling is one classic type of feedback. Heimerl et al. [25] project textual documents onto scatterplots and coordinated views such that users can re-label items while observing the changes. Simard et al. [57] propose *structured labeling* to cope with the evolution of users' decision boundaries, resulting in more consistently labeled datasets. However, user studies show that users prefer richer control over ML than simply labeling examples [2, 45, 60]. In response, many IML systems support feature selection, which we discuss in Section 2.2. Across these projects, one highly valued attribute is real-time responses to user feedback (e.g., [6, 11, 25, 54]). While rapid updates enable iteration [2], they might also encourage local interventions that are not globally beneficial—the core concern of the present work.

## 2.2 Feature Selection

*2.2.1 Forms of Feature Selection.* Interactive feature selection can take several forms. For instance, with INFUSE, Krause et al. [30] visually integrate multiple automated feature rankings and help users select features they desire. The most frequently requested [61] and supported manipulations are (1) to specify the feature space (e.g., add or delete features) and (2) to adjust feature weights based on their domain knowledge. UTOPIAN [15] interactively refines topic models as users create sets of selected keywords or adjust the meaning of a topic with keyword reweighing. FeatureInsight [10] supports "feature ideation" by visually examining sets of errors, and facilitates binary text classification through word-based feature selection. May et al. [41] help users filter redundant features. EMR VisWeb [63] enables clinical researchers to review and select word features from clinical text. DUALIST [55] updates an email classifier with features deemed positive or negative. Similarly, EluciDebug [33] supports experimentation with different feature sets and their corresponding weights. All of the works include certain forms of local interactive feedback based on stage-sensitive information. In fact, though there is no official definition for "local feature selection," we suspect most interactive feature selection fall into this category because of their iterative nature. As prior work has found that people are poor at quantitatively weighting features [14], we focus our study on the addition and removal of features.

*2.2.2 The Impact of Feature Selection.* Other projects have examined the impact of feature selection. Reunanen [53] notes the potentially misleading effect of hill-climbing cross-validation performance with exhaustive search with the *automated* sequential forward floating selection (SFFS) algorithm, and suggests simple search strategies are less prone to overfitting. We similarly contrast accuracies on development and test validation sets in a hill-climbing setting; however, our work focuses on human intervention, which as we show does not rely solely on following a model performance gradient (we discuss these differences further in Section 8).

Raghavan et al. [49, 50], on the other hand, find a positive effect of interactive feature selection in active learning, when a small amount of training data provides limited information (at most 50 documents in their case). In a similar text classification context, Raghavan et al. compare the overlap between the tokens deemed relevant by users and those ranked high by information gain (IG) to show that users are capable of identifying important token features. They then simulate a human-in-the-loop experiment to assess how inclusion of feature selection in addition to instance labeling can boost performance relative to labeling alone. Their work, however, does not consider the potential effect of model performance information on users' sequential feature selections. Moreover, as an increment to active learning, their evaluation does not include separate development and test validation sets, limiting any assessment of generalizability.

Along the lines of active learning and feature labeling, Das et al. [16] also notice that idealized feature labeling based on IG can speed up active learning and improve model performances. Their user study suggests that actual end user labeling provides less of a gain than oracle feature labels, and that some semi-supervised feature labeling algorithms perform even worse than algorithms that ignore the feature labels. Das et al. use a different scenario than ours, as their work does not consider model performance feedback. Also, they allow users to provide features not originally in the learning algorithm's data representation, granting more capability to users than just feature selection. However, their observation that end users may provide noisy feature labels helped inspire our work.

*2.2.3 Application Domain: Sentiment Analysis.* As evidenced by the above projects, interactive feature selection is often used in text classification tasks (e.g., [10, 15, 33, 55]). This is understandable, as text features (words and phrases) are naturally human understandable. We similarly base our work in text classification, specifically binary sentiment analysis: determining if the tone of a document is positive or negative [43]. As a general audience can understand the task of sentiment analysis, this domain lets us study how users balance multiple decision factors, namely prior knowledge and performance feedback (Section 2.3 below).

### 2.3 Decision Factors in IML

Prior work has identified two primary factors affecting user decision-making in IML. On one hand, researchers have observed model performance metrics to be commonly available, highly desired, and especially influential. Trivedi et al. [63] report users' desire for a performance report for the model in each iteration, and Stumpf et al. [60] notice that users paid specific attention to the extent of the accuracy variation. Amershi et al. [5] show that end users regarded leave-one out cross-validation accuracy as a quantity to maximize, and participants in Fiebrink et al.'s [19] study also report treating a high cross-validation accuracy as reliable evidence that a model was performing well.

On the other hand, the importance of users' prior knowledge has also been acknowledged. Stumpf et al. [60] categorize prior knowledge that users reply on when generating feedback into (a) knowledge of English, (b) commonsense knowledge, (c) domain-dependent knowledge, and (d) other. Lim et al. [37] observe that prior knowledge lessened participants' effort to be precise about their understanding. Our study investigates these two decision factors.

### 2.4 Performance Impacts of IML

While IML interactions have been extensively studied, the effect of these interactions on the learner requires more attention. Characterizing model performance—and in particular generalizability—is essential, especially as some researchers have found that iterative refinement does not always improve model performance. For instance, researchers note that continuously adding new training

Table 1. Comparison of Our Decision Study and Impact Study

| | Decision | Impact |
|---|---|---|
| Objective | Decision factors | Model performance |
| Participants | 100 MTurk workers | 25 ML practitioners |
| Stimuli | Controlled | Automatic |
| ML-model | Simulated | Random forest |

data can negatively impact performance [4] and that continuous labeling of training data can violate *iid* assumptions [19], which can be problematic unless the examples are carefully chosen.

In the area of feature selection, Stumpf et al. [60] suggest that users may make poor decisions: user-generated features tend to redundantly constrain or over-constrain the classifier. In a more recent study, Stumpf et al. [59] observe that IML explanations could lead to unintentional effects: detailed explanations promote over-reliance, whereas lack of explanations leads to excessive self-confidence. Kulesza et al. [33] compare feature refinement—adding, removing, and re-weighing features—with traditional instance labeling. They evaluate the performance score trajectory as subjects update models and empirically classified features into "obvious" and "subtle" buckets, and confirm that explanations in IML could help improve subjects' mental models and lead to more effective model improvement. However, they also notice that feature-based feedback may not always result in the most accurate classifier. Despite these observations, little work has investigated (1) how users generate their feedback for the ML system, and (2) what are the effects of such inputs. We seek to help fill this gap, starting with a concrete investigation in the case of sentiment analysis.

## 3 GENERAL CONTEXT AND METHOD OVERVIEW

As explained in Section 2.2.3, our study focuses on sentiment analysis for its generality, simplicity, and because most people have common-sense intuitions about the domain knowledge. We use generalization performance in order to assess the impact of user input. Cross-validated model performance measures—precision, recall, and F1 score (their harmonic mean)—are standard in both traditional and interactive ML systems. However, repeatedly testing on the same holdout across iterations can lead to overfitting. To appropriately evaluate models, we perform a three-way split of the data into training, development, and test sets [2]. We use the training set to build the ML models, the development set to provide per-iteration performance feedback, and the test set to evaluate the final model performance subsequent to interactive modification.

Our work starts with two related human-subject experiments with different aims (Table 1). The first study (the "decision study," Section 4) evaluates the factors driving users' local feature selection decisions. For our experimental factors, we focus on (1) prior knowledge, which is what users know about the question space, and (2) model behavior presented as the change in F1 score. For this study, we recruited participants from Amazon Mechanical Turk. To examine their decision processes in a controlled environment, we used a simulated ML-style interaction, rather than an actual ML system. We do this because a change of F1 score that is unpredictable or falls into too narrow a range is insufficient for studying how participants react to varying reported performance differences. Instead, we manually created strictly balanced (thus artificial) performance scores in this study. The second study (the "impact study," Section 5), focuses on how participants' feature decisions affect the performance of a real ML system. Prioritizing ecological validity, we recruited ML practitioners from our university and provided them with actual classification models, performing real-time model updates and reporting changes in F1 scores. We compare and discuss these results in Section 6.

We then leverage our experimental results to construct a set of *simulation experiments* (Section 7) that test a suite of interactive feature selection criteria at scale. A simulation-based approach enables us to assess the effects of different utility functions and feature selection strategies, as well as choices of dataset, training set size, and learning algorithm. These simulations also allow us to examine the asymptotic behavior of interactive feature selection criteria, providing more conservative estimates that extend beyond early termination due to user fatigue.

## 4 STUDY 1: DECISION STUDY ON MTURK

The decision study examines how subjects balance their prior knowledge with system-reported feedback on model performance. We hypothesized that both factors strongly influence decision making for interactive feature selection, and created a strictly controlled environment with manually selected variables to verify the following hypotheses. In order to control performance score changes, this study did not use an actual ML model; however, subjects were under the impression that the scores came from a real machine learner. To keep this distinction clear, this study refers to changes to the features of "the imagined ML model." We make the following three hypotheses:

*H1. User decisions are influenced by prior knowledge.* Studies have shown that prior knowledge biases decision making under uncertainty [23], visualization perception and interpretation [26], and understanding of vague phrases [22]. These studies lead us to hypothesize that humans will trust themselves more for feature decisions on words with stronger polarities.

*H2. User decisions are influenced by performance feedback.* Supervised learning algorithms are often explicitly designed to maximize generalization accuracy [19]. To assess performance, users likely want to view accuracy (or F1 scores) while manipulating features [63]. We hypothesized that such performance feedback affects users' decisions.

*H3. Decision-making time increases if prior knowledge and performance feedback do not agree.* Conflicting information could slow people down. Interaction delays might serve as a useful indicator in this case, showing that users are aware of the conflict between their prior knowledge and the model feedback. We therefore measure users' response time (RT) to assess this potential effect.

### 4.1 Participants

We recruited participants from Amazon's Mechanical Turk, limiting the participant pool to subjects from within the United States, with a prior task approval rating of at least 97%, and at least 1,000 approved tasks. To ensure the quality of the data, we rejected five participants post-hoc, who (1) spent less than 1.5 seconds on more than half of the questions and (2) disagreed with most participants on more than half of the questions. We collected data from a total of 100 participants.

### 4.2 Context: Sentiment of Movie Reviews

To provide a well-defined context to participants, we collected a random subset of IMDB Movie Reviews [39] as our user study dataset, and asked participants to indicate if specific words were useful for machines to predict the sentiment (positive or negative) of movie reviews. We believe this scenario is representative and intuitive (i.e., movie reviews require mostly common sense), such that participants' performances were not unduly hindered by task difficulty.

### 4.3 Procedure

The study started with an introduction, in which we explained the context and tasks: "help the machine judge movie review sentiment." To avoid biasing subjects' attention, we did not explain our focus on prior knowledge and performance feedback. To verify they understood the instructions, we asked four quiz questions. Each subject then performed 56 tasks, each judging the relevance of a possible feature. Afterwards, participants completed a brief survey, involving Likert scale [36]

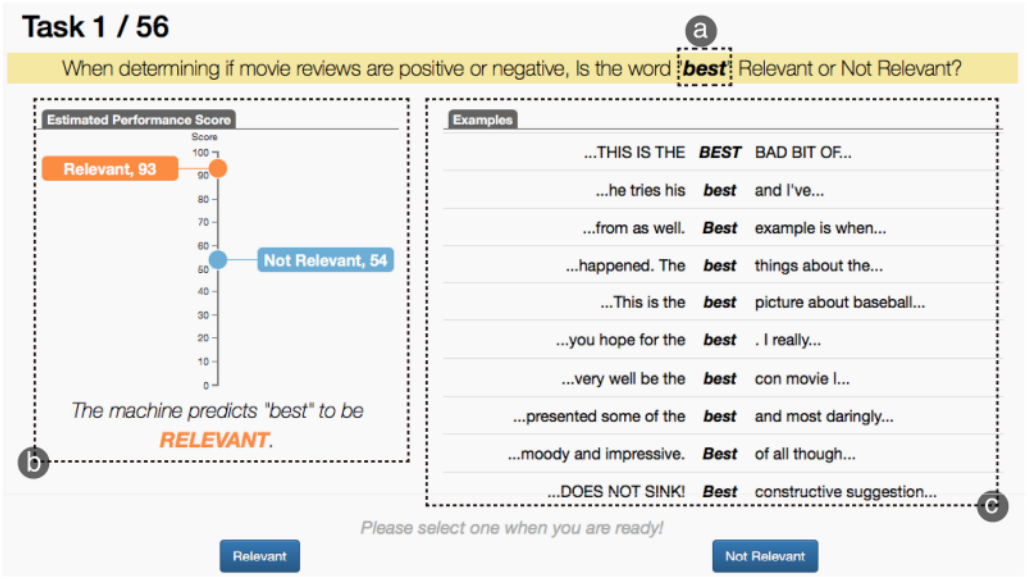


Fig. 1. The interface for the decision (Section 4) and impact studies (Section 5), providing (a) the selected word (intended to prime implicit prior knowledge), (b) performance feedback ($\Delta S = 39$), and (c) examples of the word used in the training corpus.

judgments on a 1–5 scale (See the analysis in Section 6.2). We asked the following: (1) whether subjects read the examples carefully, (2) the importance of the estimated score, (3) the accuracy of the estimated score, and (4) the importance of their prior polarity knowledge. We wished to see if their self-reflections matched their word judgments. Upon completion, each participant received $1.45 USD in compensation, which corresponds to an hourly wage of approximately $6.90/hour.

### 4.4 Task and Experimental Conditions

In each of the 56 tasks, participants were given one individual word (unigram) and asked to determine if the word is "relevant" for machines to predict sentiment or "not relevant." Corresponding to our hypotheses, we defined two variables to approximate prior knowledge and performance feedback, respectively, i.e., the *prior polarity* and the *performance score* (introduced below). To measure the impacts of both, we varied the tasks by strictly guiding the selection of the 56 words with these two variables (Section 4.5). We created a UI that provided subjects with three forms of information (Figure 1) for each word/task:

*Prior Polarity* (Figure 1(a)): Subjects' prior knowledge was primed by simply reading the word. We believe subjects weigh the general polarity of the given word when judging its relevance. We estimate polarity, $P$, using SentiWordNet scores [7]. For words with multiple part-of-speech features, we took the average of all their synset scores as the final polarity.

*Performance Score* (Figure 1(b)): We displayed the performance feedback as two scores in the range $[50, 100]$: $S_r$ (the predicted score of the imaginary model if the word is deemed relevant and included in the model) and $S_i$ (the score if the word is deemed irrelevant and excluded from the imaginary model). Participants were told the score showed how relevant or not the machine estimated a word to be, but that the scores were only estimated and could be wrong. We show both the relevant and irrelevant scores to give subjects a sense of the range of potential performances; however, the relative difference in scores is the essential signal that we want to convey. For

instance, Figure 1(b) indicates our artificial model rates word "best" to be undoubtedly useful—using it as a feature could boost the performance by 72%, from $S_i = 54$ to $S_r = 93$. This large estimated boost may induce subjects to mark "best" as a "relevant" for the imaginary model.

*Examples of Word Use* (Figure 1(c)): For context, participants were shown 10 randomly selected example reviews. Instead of showing the full reviews, which were lengthy, we displayed fragments (five words before and after the given one). This design reflects the widely accepted idea in linguistics that "a word is characterized by the company it keeps" [20]. The experiment introduction emphasized that we wished to understand the general case, not any specific examples, so we could use the words to train a machine to predict the sentiment of other reviews not yet written.

We expect the subjects to weigh these three signals when determining relevancy of a word. In Figure 1's case, subjects are highly likely to deem "best" to be relevant to movie review sentiments, as the prior polarity, performance score, and the examples all consistently suggest it. In addition, we also create cases that involve conflicting signals (explained below). Quantifying the polarity and the performance score and comparing them to users' final selection in each task, we should then be able to analyze how subjects balance this two.

### 4.5 Word Selection: Artificially Balanced

We selected a list of words to carefully balance the distribution of both prior polarities and performance scores. This procedure includes three steps as follows:

*Step 1: Sample 56 words based on estimated polarity scores.* Starting with a word feature list ranked with IG [28], we matched each word with its SentiWordNet polarity score (a continuous score in the range $[0, 1]$ that we refer as $P$) [7]. To ensure we chose words with evenly distributed polarity scores, we did not directly use the continuous $P$ values; instead, we stratified them into discrete levels $L_p$, with the strata computed via uniform steps. With $L_p$, we then manually selected a list of 56 words that were relatively highly ranked by IG, with an equal number of words in each polarity level $L_p$. As our pilot studies confirmed that the sign of the polarity did not affect relevance judgments, we broke the absolute value $|P|$ into *four levels* $L_p$: neutral (0), low (1), medium (2), and high (3). Each level includes 14 words, and includes both positive and negative polarities—For instance, $L_p = 3$ corresponds to a word that is strongly positive or negative. Note that this polarity score is only estimated (a limitation discussed in Section 8.2.1), and is not displayed in the UI. The actual in-context polarity is perceived by the subjects only by reading the word.

*Step 2: Assign artificial scores to each selected word.* We generated 56 pairs of *artificial* performance scores $(S_r, S_i)$. To best mimic a real model, all the $S_r$ and $S_i$ are randomly sampled between 50 and 100. In the process, we balanced (and thus to examine participants' reactions to) (1) the *direction* of the performance estimation (i.e., relevant or irrelevant, reflected by $sign(S_r - S_i)$ being 1 or $-1$), and (2) the *significance of the score difference* ($|\Delta S| = |S_r - S_i|$). The final range for the score difference, $S_r - S_i$, is $[-50, 50]$. Similarly, to ensure even distribution as in step 1, we stratified this performance difference range into seven levels $L_s$, from highly irrelevant ($-3$) to highly relevant (3). The direction of $S_r - S_i$ causes the levels to be doubled here.

*Step 3: Balance the polarity and the performance score.* We evenly paired the level of *polarity* and *performance* to evaluate their interactions. The final distribution includes two words for each combination of $L_p \in [0, 1, 2, 3]$ and $L_s \in [-3, -2, -1, 0, 1, 2, 3]$. Note that $L_p$ and $L_s$ are only used for sampling and pairing. We use the original continuous performance score $P$ and performance score $\Delta S$ in the following computational analysis.

As shown in Table 2, the selected words comprehensively cover both *intuitive* cases, where the polarity and performance score agree on word relevancy (e.g., "awful"), and *surprising* cases, where the two values point to different directions (e.g., "amazing"). The 56 words were independent of

Table 2. Example Words from the Artificially Generated List

| Word | $\|P\|$ | $S_r$ | $S_i$ | $\Delta S$ | $L_p$ | $L_s$ |
|---|---|---|---|---|---|---|
| awful | 0.875 | 97 | 51 | 46 | 3 | 3 |
| worst | 1.0 | 52 | 50 | 2 | 3 | 0 |
| amazing | 1.0 | 52 | 94 | −42 | 3 | −3 |
| life | 0.25 | 96 | 58 | 38 | 0 | 3 |
| prior | 0.0 | 73 | 80 | −7 | 0 | 0 |
| thing | 0.125 | 58 | 90 | −40 | 0 | −3 |

each other, and were randomly ordered for each participant. This balanced design should prevent biasing subjects toward either factor.

### 4.6 Results

We collected users' decisions and their RT for each word, as well as their self-ratings in the survey questions. Here, we analyze the impact of prior knowledge and performance scores on what participants chose, and how long it took them to choose it. We discuss the self-reflection results later, in order to compare between the decision and impact studies.

*4.6.1 Decision Strategy: Polarity > Performance.* We fit a logit mixed-effects model to analyze how participants' decisions vary with prior knowledge and performance feedback. The dependent variable was users' binary relevance judgment. Our model included as fixed effects the performance feedback $\Delta S$, the absolute continuous polarity of the prior word $|P|$, and their interaction $\Delta S \cdot |P|$. We also included a maximal random effects structure [8], with a per-subject random intercept (capturing individual decision thresholds) and per-subject random slopes for all fixed effects (capturing varied sensitivities to those factors). We normalized both the polarities and the performance scores for easier comparison.

The fitted (fixed effects) model formula for the logit is as follows:

$$I(Relevant) = \mathbf{-1.301} + \mathbf{1.061} \cdot \Delta S + \mathbf{3.464} \cdot |P| - \mathbf{0.621} \cdot \Delta S \cdot |P|.$$

The negative intercept indicates that subjects default to judging a word "irrelevant." The slope for $\Delta S$ is smaller than the intercept, indicating that performance scores alone are usually not enough to judge a word relevant. The slope for $|P|$, on the other hand, is around three times both the intercept and the slope for $\Delta S$. This suggests that prior knowledge played a more important role in participants' decision process.

The analysis supports both hypotheses H1 and H2. We observed significant effects of both performance score $\Delta S$ ($z = 8.504$, $p < 0.001$) and polarity $|P|$ ($z = 17.324$, $p < 0.001$). The interaction term was also significant ($z = -3.085$, $p = 0.002$); the negative value indicates that the predictive power of performance and polarity is maximized when the other factor is a conflicting indicator; for instance, when the system reports strong performance feedback for a neutral word, it has much more impact than similar feedback on a strongly positive word.

*4.6.2 Response Time: Conflict Slows Participants Down.* Overall, the average time to complete the decision study was $\mu = 13.63$ minutes ($\sigma = 5.807$). We analyzed RTs with a linear mixed-effects model, including the following four factors as independent variables: (1) the word order $o$, (2) the absolute polarity of a word $|P|$, (3) the performance feedback $\Delta S$, and (4) the interaction term $\Delta S \cdot |P|$. Again, we included a per-subject random intercept and random slopes for all fixed

effects. We found a significant impact of the word order; participants spent less time on each word as they advanced through the study ($F(1,5482.1) = 1417.76$, $p < 0.001$). This makes sense, as the more words they finished, the more familiar they were with the task. We also observed a significant effect of the interaction term $|P| \cdot \Delta S$ ($z = 8.580$, $p = 0.003$), indicating that participants took longer if performance and polarity provided conflicting clues, confirming hypothesis H3. This result may suggest a heuristic for identifying confusing cases where more supportive information may be helpful. We did not observe any significant effect of $|P|$ ($F(1,99.1) = 0.00$, $p = 0.999$) or $\Delta S$ ($F(1,99.3) = 0.21$, $p = 0.649$) on the RTs.

## 5 STUDY 2: IMPACT STUDY WITH ML PRACTITIONERS

We now investigate how users' feature-selection decisions affect the performance of ML models. Given the potential risks of either being over-confident in one's own knowledge or being over-reliant on performance feedback, we hypothesized that *(H4) participants' selection of features will not lead to improved model accuracy.*

We used a similar study design and the same interface (Figure 1) as the decision study, but with real ML model updates and skilled users.

### 5.1 Participants

We recruited researchers and students from our university, using email lists associated with departments like Computer Science and Engineering, where potential subjects were likely to have significant computer experience. We used a screening survey, limiting subjects to those over 18 years old with basic ML knowledge. We ended up with 25 participants who have taken one or more ML courses and/or have experience building ML models. On a 1–5 scale, subjects' average self-rating of ML expertise was $\mu = 3.24$ ($\sigma = 0.879$). We compensated subjects with $10 gift cards.

### 5.2 Study Settings: Actual Sentiment Analysis Model

To ensure ecological validity, we used a real sentiment analysis model with random forest classifiers [12, 29]. In pilot studies, we observed consistent results for a number of candidate algorithms. We then chose random forests for their speed, an essential factor for the real-time iterations required in this study.

For each participant, we randomly sampled 400 movie reviews as the model's underlying training dataset, and 100 as the development set. This is because studies that purely involve feature selection normally use 100–1,000 training data points [10, 30, 53, 60]. We used a consistent test set with 2,500 reviews—A larger dataset than the training set for testing generalizability (also seen in [61]). For training the model, we initialized our *feature space* to be all the uni-, bi-, and trigrams except extremely rare (frequency smaller than 0.005) or common (frequency larger than 0.995) ngrams. As a result, our model starts with around 6,460 features in total. We ranked all these features by their IG [28] and took the top 56 words to be the *partial feature subset* for gathering user inputs. Though selecting features based on the IG takes the risk of potentially including more relevant features than irrelevant ones, this strategy follows prior work on prioritizing the most important features for users to interact with [33, 61, 63].

We displayed these 56 words to participants in sequence. For each word displayed, we trained a model that used the entire current feature space (i.e., including the word), and one that used all features *except* for the current word. We tested the two models against the development set, and used the resulting F1 scores as the *performance score* for the "relevant" and "irrelevant" options.

Each time a participant judged a word, the feature space was updated accordingly. If the subject chose "relevant," no change was made. Otherwise, the word was excluded from the feature space, and the model was re-trained. Afterwards, the system presented the next word to the participant,

| | L_s | | | | | | |
|---|---|---|---|---|---|---|---|
| L_p | -3 | -2 | -1 | 0 | 1 | 2 | 3 |
| 0 | 2.167 | 3.750 | 2.182 | 2.391 | 2.455 | 2.375 | 2.409 |
| 1 | 1.864 | 2.348 | 1.722 | 1.810 | 1.947 | 2.091 | 1.944 |
| 2 | 2.000 | 2.773 | 1.957 | 2.250 | 1.684 | 2.182 | 3.333 |
| 3 | 2.045 | 2.909 | 2.526 | 2.609 | 2.095 | 2.636 | 2.591 |

Fig. 2. Averaged word feature count in the impact study, with respect to the polarity level $L_p$ and the performance level $L_s$.


Fig. 3. ΔF1 distribution collected from the impact study, with the red bar being the "no-change" indicator. The 95% Confidence Interval overlaps with 0, indicating no significant effect: interactive feature selection was largely a waste of time in our sentiment-analysis domain.

with the updated model used to estimate performance scores. In other words, participants made serial changes, such that prior decisions could affect the model performance seen later. Depending on how participants decided to modify the features, the updated models were trained with a varying number of features, ranging from 6,404 to 6,460.

Note that in contrast to the decision study, in which the task words were manually selected and balanced, we had no direct control over the 56 words, their display order, or performance scores—these were completely determined by the randomly selected datasets, the users' actions, and the learned models themselves. In other words, they all varied for each individual.

### 5.3 Results

As in the decision study, we collected participants' decisions and their RTs for each word, as well as their responses to the exit survey questions. We first analyzed how subjects' decisions impacted their models. We then modeled participants' decision processes and their RT using the same approach as in the decision study.

*5.3.1 Retrieving Data Distribution.* Since the training dataset was uncontrolled, we analyzed the data distribution. Figure 2 suggests that the automatically generated data also has a fairly balanced distribution. Out of the 56 words, our model suggested 25.8 words to be "irrelevant" on average ($SD = 4.37$). In these suggested "irrelevant" cases, the average performance score difference, $\Delta S$, was $-0.043$ ($SD = 0.035$); The score for relevance cases was ($\mu = 0.0491, SD = 0.040$). As for decisions, our participants removed/deemed 28.3 words ($SD = 5.06$) to be irrelevant on average, among which around 14.2 were aligned with the model suggestion.

*5.3.2 Impact: Interactive Feature Selection Can Degrade Model Performance.* We recorded the initial model $m_0$ and the final model $m_1$ (i.e., after 56 feature manipulations) for each participant. While the average performance on the *development set* went up with F1 climbing from 0.767 to 0.807, the average *test set* F1 dropped from 0.774 to 0.763. This indicates that while user input improved "local" F1 scores, it often led to poorer overall performance due to overfitting. However, as shown in Figure 3, the distribution of F1 score differences on the test set $\Delta F1_{test}$ has a fairly wide spread ($\mu = -0.011$, $\sigma = 0.019$). We computed a 95% confidence interval (CI) via bootstrapping, sampling subjects with replacement. The interval overlaps with zero, and a Wilcoxon signed-rank test ($z = 113$, $p = 0.230$) fails to reject the null hypothesis that the true mean is equal to zero.

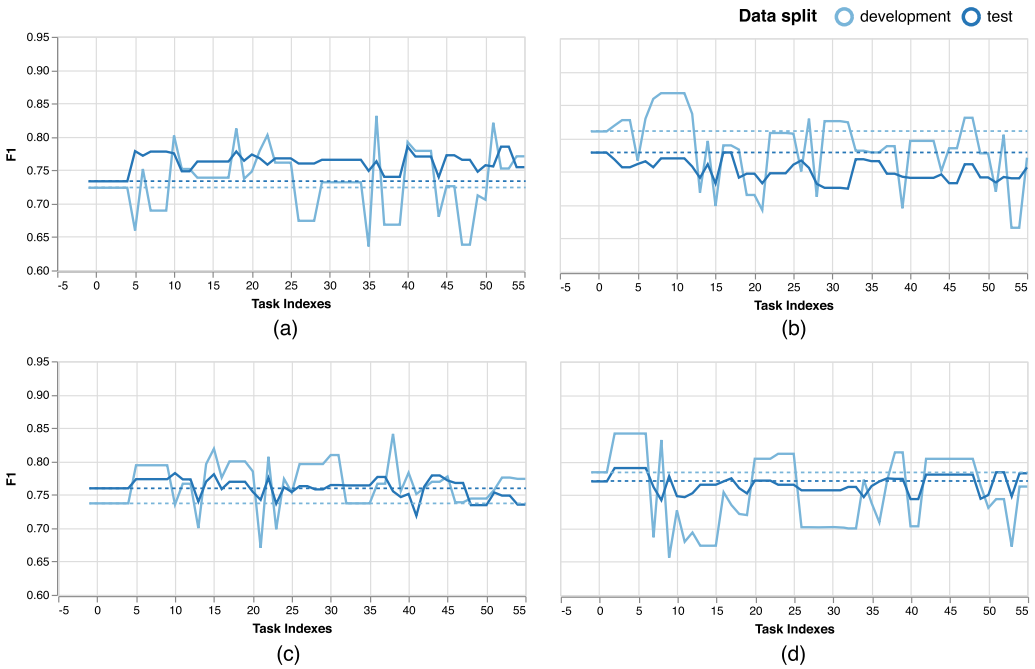


Fig. 4. Performance trajectory examples for four participants along the 56 tasks. The F1 scores oscillate for participants who modify the model to (a) have increased performance on both the development and test set, (b) have decreased performance on both sets, and (c, d) increase performance on either the development or the test set, and decreased score for the other dataset.

In other words, we find no significant effect of interactive feature selection on the final model performance (hypothesis H4).

Unfortunately, we were not able to identify key factors that lead to those $\Delta F1 > 0$ cases (word ordering, participants' choices on certain words, number of words determined as "relevant," etc.) In fact, as in Figure 4, even for individuals who successfully increased the model performance on the test set, their F1 trajectory along the 56 tasks showed no monotonicity. To the best of our knowledge, the observed widespread ΔF1 seems to be a result of users' randomness. We suspect this is because participants can hardly develop a complete understanding of the data given prior knowledge and local performance feedback, and therefore their feature updates just sway around the idealized feature distribution. Some participants might select features that fit the structure of the data by coincidence, while others were less "fortunate" and injected "mismatches between model assumptions and problem structure" [16].

*5.3.3 Decision Strategy: ML Practitioners and Turkers Act Similarly.* Though the impact study was not conducted with controlled word polarities or performance scores, we can still fit a logit mixed effects model to understand the decision strategies of ML practitioners. Interestingly, we find that the slopes and intercept for ML practitioners are very similar to those of Mechanical Turk workers, suggesting they share similar strategies:

$$I(Relevant) = \mathbf{-1.667} + \mathbf{1.099} \cdot \Delta S + \mathbf{3.270} \cdot |P| - \mathbf{0.054} \cdot \Delta S \cdot |P|.$$

The impact of $|P|$ remained strong ($z = 14.066$, $p < 0.001$). The significance of $\Delta S$ was slightly lower ($z = 2.642$, $p = 0.008$). We did not observe a significant effect for the interaction term.

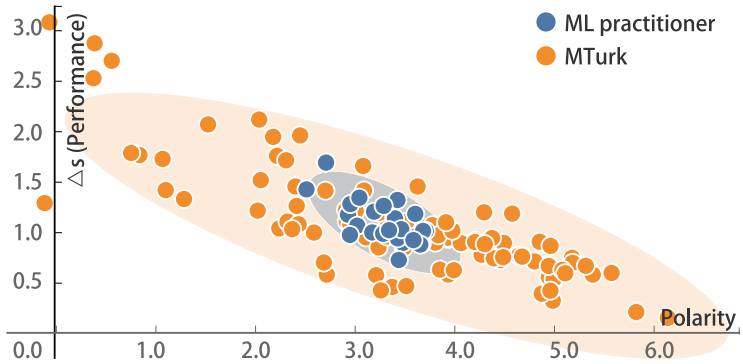


Fig. 5. Scatterplot contrasting the per-subject random slope terms for polarity $|P|$ and model score $\Delta S$ with one dot for each subject and covariance ellipses. We observe both groups have similar behavior, but ML practitioners make more consistent decisions.

*5.3.4 Response Time: Similar Pattern, Slower Overall.* The average time $\mu$ to complete the impact study was 17.31 minutes ($\sigma = 5.681$). Note that this is around 4 minutes longer than in the decision study. The differences in the average completion time could result from either the participant differences (e.g., ML practitioners completed the tasks more carefully), or that the impact study was designed to take longer by nature, as participants had to wait briefly for the model to retrain, after each iteration. A linear mixed effects model revealed similar patterns: completion time is significantly correlated with the word order ($F(1,1358.74) = 53.494$, $p < 0.001$) and the interaction term $\Delta S \cdot |P|$ ($F(1,1057.29) = 4.181$, $p = 0.041$).

## 6 CROSS-STUDY COMPARISON

With both studies at hand, we analyze the differences between the MTurk workers and ML practitioners. We first inspect inter-subject differences and then analyze the exit survey results.

### 6.1 Decisions: ML Practitioners Are More Consistent

Recall that our mixed effects models include per-subject random slopes for all fixed effects. To inspect inter-subject variability, we examine the estimated per-subject random slope terms in our models to inspect how consistent the users in different groups behave. In specific, we adjusted the slopes from the fitted models (in Sections 4.6.1 and 5.3.3) for each subject according to his/her random slope, so to recover the individual behaviors. Figure 5 shows a scatterplot of coefficients (adjusted per-subject) for polarity $|P|$ versus performance feedback $\Delta S$, with the covariance ellipse for 95% CI overlaid.

The plot shows to what degree individuals more strongly weigh these factors than the average value, and indicates the level of variation in each subject population. For instance, while in Section 4.6.1 we find "the slope for $|P|$ is around three times the slope for $\Delta S$" (referring to the dense distribution of orange dots around (3.464, 1.061)), those orange dots in Figure 5's upper left corner reveal that there are also MTurk users who, in contrast to the general trend, consider the estimated model performance feedback to be more essential. The covariance ellipse shows that ML practitioners (blue dots) are more compactly clustered, showing greater strategical consistency, whereas MTurk workers (orange dots) exhibit higher variance in inter-subject differences.

### 6.2 Self-Reflection: Polarity Preferred to Score

Finally, we contrasted the self-reflections from MTurk workers and ML practitioners (Figure 6). Most of our participants in both studies reported that they read the examples carefully ($\mu = 4.71$,

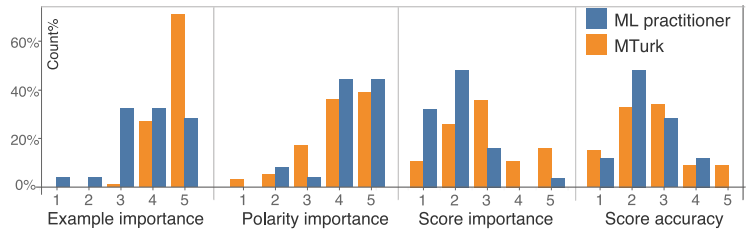


Fig. 6. Participant self-reflections: both populations reported that they read examples carefully, and cared more about polarities. Compared to MTurk users, ML practitioners believed the system score feedback was less important and less accurate.

$\sigma = 0.478$ for the decision study; $\mu = 3.76$, $\sigma = 0.880$ for the impact study). Prior knowledge of word polarity ($\mu = 4.03$, $\sigma = 1.020$) was rated more important than the performance feedback ($\mu = 3.24, \sigma = 1.124$) for MTurk workers. This result may be due to subjects' belief that the performance feedback was not sufficiently accurate ($\mu = 2.95$, $\sigma = 1.209$): we observe a positive correlation between how important participants think the performance feedback is, and how accurate it is ($r = 0.626, p < 0.001$). The self-reported importance bias is even larger for ML practitioners: ($\mu = 4.24$, $\sigma = 0.879$) for polarity importance versus ($\mu = 1.96$, $\sigma = 0.934$) for score importance.

## 7 SIMULATION STUDIES

Using our model of decision making factors from our user studies, we look beyond the "word relevance" context for movie reviews, and use the model to simulate interactive updates in additional settings. Our primary goal is to enhance generalizability by extensively testing if our user study results hold for multiple conditions, without requiring extensive (and intractable) additional user studies. Our simulations include various (1) *utility functions*, which reflect how users weigh different factors in feature selection, to see which local decisions are most aligned with the global model performance improvement, (2) *feature selection strategies* to see if alternative strategies can be more beneficial, (3) *datasets with different characteristics and sizes* to assess potential effects of dataset and training set size, and (4) *different algorithm families with varying regularization settings* to see if certain classification algorithms are more suitable for interactive feature selection than others.

A secondary goal of our simulations is to understand model performance impacts in the asymptotic case. Real user studies need to be of reasonable duration to avoid fatigue and not exceed a maximum number of features that a user is willing (or able) to consider. Simulating users making consistently qualified choices until convergence can help verify if observed performance detriments result from early termination, or from more inherent issues.

### 7.1 Workflow and Context

We simulate a user that iteratively refines a sentiment analysis model in a greedy manner. Our simulation runs as follows:

1. *Model Initiation.* Given a dataset, we randomly sample half of the data to use as the test set. This test set is used across all the simulation processes to ensure fair comparisons. We then sample a certain number of data points (pre-determined as described below), and split them into training and development sets. We build the initial model based on the training set.

2. *Iterative Feature Selection.* We start the iterative refinement process with an initial feature set derived from the initial model. The feature set is ranked by IG [28]. We then make feature selection decisions in order, based on one of three *utility functions*. In each iteration, we edit features

Table 3. Conditions Tested in the Simulation

| | Count | Condition |
|---|---|---|
| | 100 | Repeated runs |
| × | 3 | Utility functions |
| × | 3 | Feature selection strategies |
| × | 3 | Datasets |
| × | 3 | Sample size for training and development set |
| × | 3 | Classification algorithms |
| × | 4 | Regularization settings |
| = | 97,200 | Runs |

following some *strategy*, and examine if the change should be accepted based on the *utility function*. We keep running the iteration to convergence: it is not stopped until there are 1,000 consecutive changes rejected (i.e., actions taken that do not satisfy the utility function's requirement). We also record the model's performance against the test set with each iteration.

*3. Evaluation.* As in the impact study, we compare the initial and the final F1 scores to assess performance changes (ΔF1). Negative ΔF1 values indicate decreases in model performance.

### 7.2 Experimental Conditions and Hypotheses

We run simulations across a variety of parameterizations (Table 3). We conduct 100 simulation runs for each condition to address potential sample bias across training/development/test set splits.

*7.2.1 Utility Functions for Accepting or Rejecting Changes.* Our user studies found that both prior knowledge and performance feedback are important, and that humans tend to balance them. In response, we simulate three different utility functions, to test both the two extreme cases where users hill climb solely based on prior knowledge or model performance, and a balanced case using the coefficient weights from our mixed effects models.

Following the notions in the user study, the two extreme cases are (1) *Pure polarity* $|P|$ (*Polarity*), where highly sentimental words are preferred, and (2) *Pure performance score* $\Delta S$ (*Score*), where the model's F1 score performance changes positively on the development set. For (3) *Balanced mixture* (*Mixture*), we simulate user decisions with our logistic model trained on ML practitioners. Because the interaction term did not show any significance, we retrain the model omitting that term. The resulting equation for the log-odds is $-1.676 + 1.070 \cdot \Delta S + 3.280 \cdot |P|$. We hypothesized that *(H5) The two extreme (pure) cases will cause more harm than the balanced case*, as they directly correspond to the over-confidence and over-reliance traps.

*7.2.2 Feature Selection Strategies.* We devised three feature selection strategies, informed by conversations with ML practitioners in our university, to evaluate if certain strategies are more beneficial than others:

*Deleting features* (*Delete*). Starting with all the unigrams, bigrams, and trigrams, users deduct features from the full feature space to prevent models from being overly constrained.

*Similarity-based feature replacement* (*Replace*). Starting with unigrams, users replace one selected unigram with a set of bigrams and trigrams that contain it to find the best level of detail of certain features. For instance, compared to the single verb "destroy," multi-word phrases such as "destroy terrorism" and "destroy the movie" convey clearer positive and negative sentiments, respectively. While "similarity" might mean *semantically similar* (e.g., synonym) or *structurally related* (e.g.,

Table 4. Distributions of the Tested Datasets

| Data | #Total | #Pos | #Neg | #Avg. word |
|---|---|---|---|---|
| IMDB | 12,500 | 6188 | 6312 | 227.78 |
| Sentiment140 | 25,000 | 12500 | 12500 | 14.18 |
| Enron | 16,859 | 4,214 | 12,644 | 352.40 |

neighbors, adjectives related to a noun), our similarity definition based on word containment keeps the original feature and the replacing feature set tightly connected.

*Document-oriented feature refinement* (*Doc*). In addition to examining features, we have noticed that practitioners sometimes focus on fixing a particular document. Our third strategy simulates this focus. The simulated user first ranks misclassified documents based on how misclassified they are. This is computed based on the documents' classification probability being positive $P_p$ and their gold label $l$ (0 for negative or 1 for positive): $|P_p - l|$. For each document, features are ranked by their IG. Starting with the most misclassified document, the simulated user replaces features that push the document toward the gold label, deleting those causing misclassification, until the document's label is corrected. All the changes made are combined into a set that is either kept or deleted as a whole based on the development set change. We hypothesized that *(H6) There is a significant difference between the three strategies* with *Doc* being the worst and *Replace* the best. We suspected the *Doc* strategy would underperform, due to overfitting from fixing specific documents, but we were optimistic about *Replace*, which has the potential to improve the model by providing more nuanced features.

*7.2.3 Data Selection. Dataset.* We simulate interactive feature selection across three different datasets (Table 4): IMDB Movie Reviews for movie review sentiment analysis [39], Sentiment140 for Twitter sentiment analysis [21], and a subset of Enron-Spam for email spam identification [42]. These datasets exhibit varied characteristics: the movie review is the most balanced dataset, the Twitter data has high sparsity, and the email spam data has unbalanced positive and negative classes. We hypothesized that *(H7) the pitfalls observed in the user study hold for datasets with different distributions and content types.*

*Dataset size.* We randomly sample $n = 500, 1{,}000,$ and $5{,}000$ data points from each dataset to form a subset, and then break them into training and development sets using a ratio of 4 : 1. As a larger training set could potentially lead to a larger feature space, we hypothesized that *(H8) the effect of feature selection decreases as the size of the training set increases.* As mentioned in Section 5.2, this sample range follows the conventions in papers that primarily focus on feature selection. We discuss a pilot simulation with a much smaller training dataset in Section 7.3.2.

*7.2.4 Algorithm Settings. Algorithm selection.* We run the simulations across three commonly used classification algorithms: Logistic Regression, SVM, and random forest, using the implementations within Scikit-Learn [46]. These belong to different algorithm families: Logistic Regression emphasizes linearity, SVM introduces nonlinearity via kernels (*rbf* kernel in our case), and Random Forest relies on Ensemble Learning to form the best model. These variations let us see if certain algorithms are especially susceptible to overfitting due to feature selection.

*Regularization.* We additionally examine four levels of regularization. For Logistic Regression and SVM, we use the L2-norm, with the *inverse of regularization strength* being $100, 1, 0.01, 0.0001$ (i.e., smaller values specify stronger regularization). For Random Forest, we set the minimum number of samples required to split an internal node as $100\%, 85\%, 70\%,$ and $55\%$ of the total number of input data to generate both full trees and pruned trees. This lets us see if aggressive regularization can rectify suboptimal user inputs, and thereby recover models from local optima.

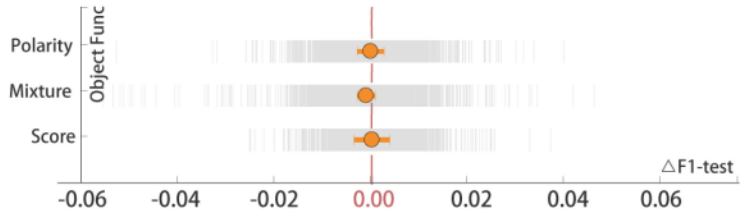


Fig. 7. ΔF1 distributions for the three utility functions, with 95% confidence intervals of the mean. The small CIs overlap with 0, indicating no reliable net effect for any utility functions.

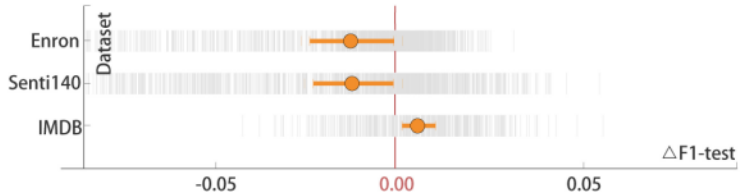


Fig. 8. ΔF1 distributions and 95% CI of the mean for three datasets. While the degree of impact varies, models for all three datasets can degrade with human inputs.

Our hypothesis for the algorithm settings was that *(H9) the pitfalls observed in the user study hold for different algorithm types and regularization parameters.*

### 7.3 Results

We seek to understand the impacts of each simulated condition on the resulting models' F1 scores. Though we examined the absolute final F1 score for the resulting models, for simplicity we report these values only for the *dataset size* and *regularization* conditions, as their results are highly related to our discussion on ΔF1. The other absolute F1 score distributions are, in contrast, less worthy of specific discussion (utility functions yield similar effects and therefore render the absolute F1 less interesting, and the absolute F1 for different datasets are not directly comparable).

We evaluate each simulation condition by computing the 95% CI of the average ΔF1 via bootstrapping, and examining if the central tendency significantly differs from zero using the Wilcoxon Signed-Rank test [67]. In addition, where applicable, we examine the differences between the ΔF1 distribution resulting from different conditions using the Kolmogorov–Smirnov test [40].

*7.3.1 Overall: Interactive Feature Selection is Not Beneficial.* Wilcoxon tests on the Polarity ($z = 1.951 \cdot 10^6, p = 0.118$), Score ($z = 1.604 \cdot 10^6, p = 0.345$), and Mixture ($z = 1.785 \cdot 10^6, p = 0.144$) utility function indicate that no net effect is discovered, failing to confirm H5. This can also be observed in Figure 7. Though all the three utility functions lead to a spread of ΔF1, the CI, while being considerably small, overlaps with 0. Despite the seemingly similar shape, pairwise KS-tests indicate that the utility functions do not result in the same distributions, with all $p$-values $p < 0.001$: $K(\text{Polarity}, \text{Score}) = 0.039$, $K(\text{Polarity}, \text{Mixture}) = 0.0556$, and $K(\text{Score}, \text{Mixture}) = 0.081$.

We did not find any significant differences between the three feature selection strategies, contrary to H6. Wilcoxon test results are ($z = 1.867 \cdot 10^5, p = 0.452$) for Delete, ($z = 1.359 \cdot 10^5, p = 0.241$) for Replace, and ($z = 1.667 \cdot 10^5, p = 0.757$) for Doc.

*7.3.2 Dataset and Size: Interactive Feature Selection Hurts by Differing Degrees.* Data-wise, we notice that all three datasets tested suffer from a potential loss of model performance (Figure 8). Though our impact study found that user input decreased average classifier performance for IMDB,

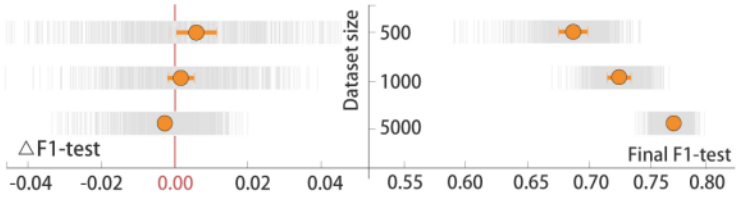


Fig. 9. Distributions and 95% CIs of the mean for (1) ΔF1 and (2) the final F1 for the converged model with respect to the three dataset sizes. CI for ΔF1 moves towards negative as the datasize grows, indicating human inputs are less desired for models with adequate training samples.

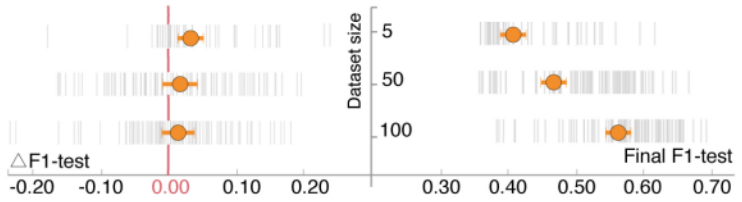


Fig. 10. Distributions and 95% CIs of the mean for (1) ΔF1 and (2) the final F1 for the converged model with respect to the three dataset sizes *used in a pilot simulation*. We observe similar trends as in Figure 9, only that the ΔF1 distribute more sparsely (note the scale), and the final F1 decreases over all.

our simulation finds a small positive improvement ($z = 3.103 \cdot 10^4, p = 0.044$ from Wilcoxon test). However, both the Enron dataset ($z = 1.540 \cdot 10^5, p = 0.002$) and the sparse Sentiment140 dataset ($z = 1.562 \cdot 10^5, p = 0.004$) exhibit negative performance changes. Again, pairwise KS-tests reveal significant differences for the ΔF1 distributions. We therefore confirm hypothesis H7. More experiments are needed to examine specific correlations between dataset characteristics and the effectiveness of interactive feature selection.

We also see that models with larger training sets are less affected by feature selection decisions, confirming H8. The variance of ΔF1 decreases as the dataset size grows. The variance for $n = 500$ ($\sigma^2 = 3.745 \cdot 10^{-4}$) doubles the variance for $n = 1{,}000$ ($\sigma^2 = 1.794 \cdot 10^{-4}$), and is roughly a factor of four larger than $n = 5{,}000$ ($\sigma^2 = 8.558 \cdot 10^{-5}$).

We further observe that feature selection is more likely to degrade the model as the training set size grows. In Figure 9, the CI for data size $n = 500$ falls in the positive range ($z = 1.206 \cdot 10^5, p < 0.001$), meaning interacting with models fit to limited training sets are more likely to be helpful. In contrast, the smaller CI for $n = 5{,}000$ distributes strictly in the negative range ($z = 1.626 \cdot 10^5, p < 0.001$), which confirms that manipulations can frequently hurt models that have seen comparatively more adequate information. The $n = 1{,}000$ condition sits in between ($z = 1.201 \cdot 10^5, p = 0.359$). Moreover, from the distribution of final F1 in Figure 9, we observe that the impact of feature selection on the resulting F1 is weak relative to dataset size. This result conforms with the common belief that more training data yields more stable models with better performance, and indicates that increasing labeling additional data may be a more effective use of users' time than low-level feature selection.

We observed similar trends in a pilot simulation where we tested models with 5, 50, and 100 training data, as papers that involve labelings of both *training data* and *feature* usually demonstrate their usefulness in the scenario where the initial labeled data is extremely limited. As in Figure 10, ΔF1 distribute much more sparsely with these data sizes; The CI for data size $n = 5$ is strictly in the positive range. However, without the *labeling* power, models' absolute performance are around

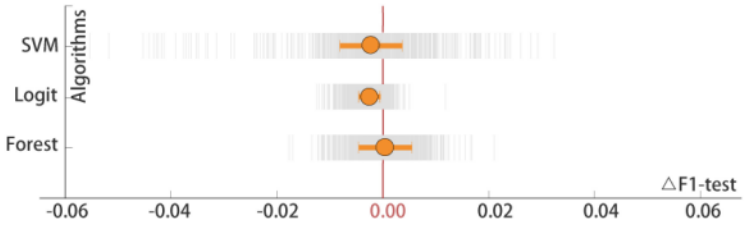


Fig. 11. ΔF1 distributions and 95% CIs of the mean for three ML algorithms. Logistic Regression yields the least variance.

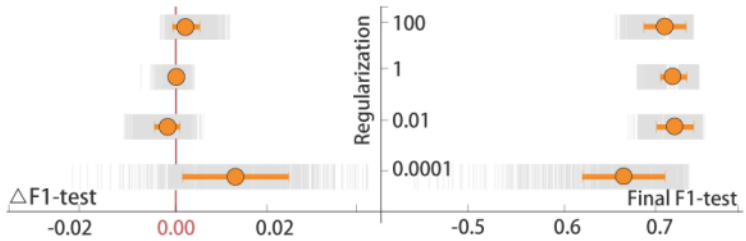


Fig. 12. Distributions and 95% CIs of the mean for (1) ΔF1 and (2) the final F1 for the converged model with respect to the four regularization terms. While aggressive regularization lifts ΔF1, the overall model performance is still low, likely due to underfitting.

random guess, and can go as low as 0.35. We deemed these sizes to be too small for a feature selection context, and therefore abandoned these sampling numbers.

*7.3.3 Algorithm: The More Expressive, the More Manipulable.* Across algorithms (Figure 11), we observe that simulations using Logistic Regression lead to more concentrated outputs ($\sigma^2 = 9.310 \cdot 10^{-5}$ is one third and one fifth to Random Forest and SVM respectively). KS-tests confirm the output of logistic regression is significantly different from SVM ($K = 0.246, p < 0.001$) and random forest ($K = 0.187, p < 0.001$). Unsurprisingly, it appears that more expressive algorithms like SVM and Random Forest are more easily affected by feature selection decisions. Furthermore, Wilcoxon tests indicate a significant negative effect on ΔF1 for Logistic Regression ($z = 1.453 \cdot 10^5, p < 0.01$). We do not see significant effects (positive or negative) for Random Forest ($z = 1.108 \cdot 10^5, p = 0.784$) or SVM ($z = 1.055 \cdot 10^5, p = 0.165$).

*7.3.4 Regularization: Larger Improvement, Worse Absolute Score.* Interestingly, regularization, a common practice to avoid overfitting, cannot effectively counteract the negative effects of feature selection. As seen in Figure 12, extremely aggressive regularization $r = 0.0001$ can shift the ΔF1 in a positive direction ($z = 2.778 \cdot 10^4, p < 0.001$). However, a comparison with the final F1 reveals that $r = 0.0001$ results in model underfitting, with diminished absolute performance.

Overall, our analysis in the above two sections confirmed H9 that our observed pitfall holds for different choice of algorithms and regularization parameters.

## 8 DISCUSSION

In this section, we first summarize our findings, and compare our work to prior studies that either encourage or discourage the use of interactive feature selection. We then discuss limitations of our study design and summarize the implications of our study.

### 8.1 Result Summary: Extend Findings in Prior Work, A Baseline for Future Work

In the context of sentiment analysis, our work studies the potential impact of interactive feature selection with a combination of human-subject experiments and computational simulations. Our experiment finds that (1) users judged feature relevance based on both their prior knowledge and model performance feedback, tending to weight prior knowledge more strongly, and (2) ML practitioners degraded model performance via local low-level feature manipulation. Based on models derived from participants' judgments, we simulated scenarios in which users build classifiers with different algorithms (regularized to various degrees) across multiple datasets with varying dataset sizes, subject to different feature selection strategies and decision functions. Our simulation finds that, in most cases, low-level feature manipulation based on local model performance and users' prior belief cannot effectively improve the model—at least not in the case of sentiment analysis.

Comparing to Reunanen [53], the pure F1 hill-climbing condition in our simulation study partially replicates their results in a different context, and corroborates the lack of benefit of feature selection via step-wise performance hill climbing. In addition, our work extends prior work by (importantly) involving users who could follow their own intuition, often in conflict with performance scores. Interestingly, we find that human intuition does not do better (or worse) than such automated methods. However, the overfitting problem is arguably even more damaging in the interactive context, as it wastes human time, not just computing time.

Comparing to prior user studies for IML systems,which provided conflicting evidence for IML, our work confirms the observation that human inputs may not improve models [2, 16, 61]. We also go one step further and show that these cases may be quite common. Frequency-wise, instead of reporting occasional decrement of model performances, our study shows that intuitive (yet naïve) feature selection has a very high chance of hurting the model. Condition-wise, throughout our studies, we observe that neither ML practitioners' input (Section 5) nor simulated utility functions (Section 7) improve the learned models. In other words, users' local interactions—based on a specific feature and the current status of the model—often fail to improve the global picture. We hypothesize that IML tasks other than basic feature inclusion/exclusion decisions might also suffer from similar shortcomings, particularly if they are similarly rooted in localized, stepwise decisions as part of a model-fitting process.

As we discuss below in Section 8.2.2, our studies do not cover all the possible variations of IML—different tasks, systems that provide feature distribution information, stable models that partially accept user inputs, and so on. We based our studies on a representative setting, such that it could serve as a baseline for the future work. Starting here, more in-depth studies can be conducted to do a "hyper-parameter" search for IML. Ideally, if we extend the study of IML impacts to other potentially beneficial elements, we should be able to characterize its strengths and weaknesses, and make statements like "certain forms of IML are likely to lead to positive outcomes under specific contexts." Until then, we wish our observations that "simple local interventions are often a waste of the users' time" can encourage more cautious use of IML systems.

### 8.2 Study Limitations

Our study has the following two main limitations: (1) our settings do not perfectly mimic IML feature selection practices in the wild, and (2) our settings seek to provide the most basic and general building blocks for IML, ignoring potential benefits from additional clues.

*8.2.1 Approximation of the Real World.* In part due to the tradeoff between being "real" and being "feasible," we chose to simplify the real-world condition, and used a simple, binary learning task, and an approximate model of users' domain knowledge.

*Users' Polarity vs. Scored Polarity: Only an Approximation.* We used SentiWordNet to quantify polarities in both the user study and the simulation. However, the dictionary scores words independent of the context in which they are used. For example, we observed "Oscar" to be a neutral word in WordSentiNet. However, "Oscar" can be highly positive in the context of movie reviews. The mismatch is further exaggerated in the cases of bi- and tri-grams, since we estimated polarity of these with the average of the associated unigrams.

*Users' Own Modeling Efforts vs. An Artificial Task.* We were surprised to see that ML practitioners tended to discount performance feedback, especially given prior work that reports subjects valuing it greatly [10, 63]. We wonder if user behavior might be different if the subjects "owned" the learning problem and so cared more about actual performance. In the context of an artificial experiment, participants may instead be more likely to behave as they think they "ought" to. In this work, we chose to trade personalized factors for better comparability. An alternative that embraces individual motivations would be to observe user interactions with their own data, task, and model. A follow-up study could investigate this further. However, this level of personalization could make the results difficult to generalize.

*Users' Context vs. Experimental Context.* Our work reflects only a small slice of practical IML application scenarios. For the sake of feasibility, our work is strictly constrained both task-wise (sentiment analysis via binary classification) and feature-wise (n-grams). In the real world, modelers may select novel classes of features within a much larger task space (image recognition, machine translation, etc.). For example, for tasks in which humans' domain knowledge is not as tangible as textual polarities (e.g., examining image metadata), we might expect users to assign greater weight to model feedback and less to their domain intuition.

*8.2.2 Focus on Straightforward Interactions between Models and Users.* Our work is conducted in a scenario where the users and the models interact naïvely. From the user point of view, the only feedback they receive are the performance score and contextual examples. In other words, we set the baseline in a complete "blackbox" setting, which means users may develop subjective and incomplete mental models [64] of how the learning system operates, leading to defective utility functions. Such mental models can potentially be rectified with additional explanations [19, 33], which may then alter the feedback these users provide to an ML system. For instance, when evaluating the effectiveness of an explanatory debugging system, Kulesza et al. [33] saw generally improving F1 scores as participants provided more feedback.

On the other hand, the model is also updated strictly following users' binary decisions on each word. Such absolute trust on user performance is not very fault tolerant, and the model performance may well suffer when users misbehave. It is possible that regarding user feedback as soft constraints rather than hard commands can help rectify potential user biases. In this vein, Trivedi et al. [63] allow batched feature feedback to avoid collecting conflicting feedback from users. However, naïve constraints can still lead to suboptimal values during training. In their experiment in which users provided constraint-based feedback, Stumpf et al. [60] noticed that the hardness of the constraints played an essential role, and that future research should reduce the potential redundancy of user feedback and models' learned results, and should prevent over-constraining. Though rarely seen in feature selection papers, prior work on feature labeling has observed that sophisticated algorithms can make effective use of modified features [16]. For instance, instead of directly modifying the feature space, Raghavan and Allan [48] created additional training data with newly labeled features.

To generalize our finding to more transparent IML systems, additional work on (1) how different kinds of explanations (feature distribution, etc.) impact users' utility functions, and (2) how

different mechanisms for incorporating user feedback affect the model performance changes will be needed.

### 8.3 Implications

We wish to make clear that we are not trying to dismiss the potential benefits of IML, nor to levy a general criticism ("IML is wrong"). Rather, we hope our work will serve as a step toward better characterizing strengths and weaknesses of different types of user interactions in IML systems, in order to better focus productive user effort. In this section, we consider three areas where IML may be more likely to lead to positive outcomes.

*8.3.1 Emphasize Best Practices from Traditional ML.* Our studies leveraged two common practices from traditional ML. First, we split the dataset into training, development, and test sets to separate development feedback and final performance testing. Unfortunately, we observed users can overfit to the given development set, leading to decreased model performance on the test set. This effect is expected, as users are repeatedly testing their (iterated) models on one single development set. Even worse, some IML systems fail to support any development set [10, 33], leading users to overfit on their test data, without any indication of lost generalizability. Studies dealing with overfitting in traditional ML can be helpful here. For instance, perhaps IML systems could incorporate a reusable holdout set [17], which uses the idea of differential privacy to safely run tests on a development set multiple times for validation.

Second, we tested effects due to regularization in our simulations. In our study, aggressive regularization pruned features such that the final model performed uncompetitively, with or without user input. However, future work might explore "appropriately" pruning poor inclusion decisions by users by tuning the regularization term to the right level. More generally, selecting an appropriate level of regularization is itself a potentially valuable IML task.

*8.3.2 Change the Scope and Interpretation of User Input.* Our work suggests local "tuning" manipulations may be harmful to model performance and prone to overfitting. A natural step forward would be to design interaction strategies that make better use of user input. We review two potential improvements here.

First, IML systems could promote more global and principled changes to prevent local oscillation. In the case of feature manipulation, instead of bag-of-word feature selection or reweighting (as in current IML systems), expanding the feature space with alternative categories of features can be more beneficial. Consider practitioners building a classifier to determine if a textual document describes "wildcats": instead of manually adding a list of infrequent nouns with names of species, one might benefit from the semantic similarity provided by word vector space embeddings, which finds words like "tiger" and "leopard" to be highly similar. However, these kinds of conceptual moves are currently the purview of ML experts and involve writing feature engineering code. Future studies examining the broader feature engineering space are needed, such that we can better understand how to design IML systems that support such manipulations.

Also, IML systems could treat users' local decisions with skepticism, interpreting user input as uncertain or "noisy" observations. For example, prior work on Snorkel [51] demonstrates that modeling potentially low-quality labeling functions can help to train high-quality end models. Similar approaches that regard user input as suggestions as opposed to hard constraints (a design limitation we discussed in Section 8.2.2) can strive for more effective improvements. A possible first step in this direction would be to automatically backtrack to previous stages and reweigh users' conflicting feedback.

*8.3.3 Enable Assessment of Training Data and Model Performance.* Beyond improving the iterative loop of model fitting, an alternative is to focus on validating the inputs and outputs of the process. Data validation can help spot biases in what the model is learning from and what it is tested on. This approach can help addresses issues that are not fixable by model iteration alone (e.g., ML fairness [9, 24] and accountability [35]) and may dissuade practitioners from making uninformed or unnecessary structural changes to models. Both assessment of training data and model error analysis seem promising. Tools for inspecting training data like Facets [62] and Flipper [66] can aid in inspecting data quality and representativeness. Going beyond standard performance feedback (F1, accuracy, etc.) into more thorough failure analysis is also helpful. Creating structured labels [13, 32] for errors in models, or hierarchically structuring their relations based on possible causes, can reflect model strengths and weaknesses on different data segments. Such processes should allow user interaction to provide more stable and reusable contributions via data correction or augmentation. Moreover, input and output validation are applicable across a broad class of modeling approaches, including the neural networks now common in practice.

## 9 CONCLUSION

This article investigates the impact of local interactions in IML in the context of low-level feature selection for text classification. With a combination of human-subject experiments and simulations, we examined the decisions users make and their impact on model performance. We observed through the context of sentiment analysis that local interactive feature selection, though supported and advocated in IML research, is not beneficial as one would hope. On average, manual tuning of unigram representations does not improve classifier performance, and in many cases degrades it. As a result, interactive tools might encourage users to perform futile actions without actually improving their models. We suspect that other IML tasks, beyond feature selection, might suffer from similar issues: if their interactions are based largely on stepwise decisions within local contexts, users may fail to globally improve their models. Though conducted in a simple binary text-classification context, our results sound a cautionary note and suggest priorities for subsequent study. Additional avenues for future work include better means of enshrining ML best practices in IML tools, the design of interfaces for more expressive high-level feature specification, and enhanced data triage and error analysis capabilities to better identify areas for improvement.

## ACKNOWLEDGMENTS

We gratefully thank our colleagues Gagan Bansal, Quanze Chen, and Yang Liu for their helpful comments. We also appreciate the valuable input from our user study participants, and the constructive comments from the anonymous reviewers.

## REFERENCES

[1] Bilal Alsallakh, Allan Hanbury, Helwig Hauser, Silvia Miksch, and Andreas Rauber. 2014. Visual methods for analyzing probabilistic classification data. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 1703–1712.

[2] 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 (2014), 105–120.

[3] Saleema Amershi, Max Chickering, Steven M. Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. 2015. Modeltracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 337–346.

[4] Saleema Amershi, James Fogarty, Ashish Kapoor, and Desney Tan. 2009. Overview based example selection in end user interactive concept learning. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology*. ACM, 247–256.

[5] Saleema Amershi, James Fogarty, Ashish Kapoor, and Desney Tan. 2010. Examining multiple potential models in end-user interactive concept learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1357–1360.
[6] Saleema Amershi, James Fogarty, and Daniel Weld. 2012. Regroup: Interactive machine learning for on-demand group creation in social networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 21–30.
[7] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC'10)*, Vol. 10. 2200–2204.
[8] Dale J. Barr, Roger Levy, Christoph Scheepers, and Harry J. Tily. 2013. Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language* 68, 3 (2013), 255–278.
[9] Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam T. Kalai. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In *Advances in Neural Information Processing Systems*. 4349–4357.
[10] 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 *IEEE Conference on Visual Analytics Science and Technology (VAST'15)*. IEEE, 105–112.
[11] Eli T. Brown, Jingjing Liu, Carla E. Brodley, and Remco Chang. 2012. Dis-function: Learning distance functions interactively. In *IEEE Conference on Visual Analytics Science and Technology (VAST'12)*. IEEE, 83–92.
[12] Rich Caruana and Alexandru Niculescu-Mizil. 2006. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning*. ACM, 161–168.
[13] 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*. ACM, 2334–2346.
[14] Justin Cheng and Michael S. Bernstein. 2015. Flock: Hybrid crowd-machine learning classifiers. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. ACM, 600–611.
[15] Jaegul Choo, Changhyun Lee, Chandan K. Reddy, and Haesun Park. 2013. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 1992–2001.
[16] Shubhomoy Das, Travis Moore, Weng-Keen Wong, Simone Stumpf, Ian Oberst, Kevin McIntosh, and Margaret Burnett. 2013. End-user feature labeling: Supervised and semi-supervised approaches based on locally-weighted logistic regression. *Artificial Intelligence* 204 (2013), 56–74.
[17] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. 2015. The reusable holdout: Preserving validity in adaptive data analysis. *Science* 349, 6248 (2015), 636–638.
[18] Jerry Alan Fails and Dan R. Olsen Jr. 2003. Interactive machine learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*. ACM, 39–45.
[19] Rebecca Fiebrink, Perry R. Cook, and Dan Trueman. 2011. Human model evaluation in interactive supervised learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 147–156.
[20] John R. Firth. 1957. A synopsis of linguistic theory, 1930–1955. In *Studies in Linguistic Analysis*. Blackwell, Oxford.
[21] Alec Go, Richa Bhayani, and Lei Huang. 2009. *Twitter Sentiment Classification Using Distant Supervision*. CS224N Project Report. Stanford.
[22] Shu-Ping Gong and Kathleen Ahrens. 2011. The prior knowledge effect on the processing of vague discourse in Mandarin Chinese. In *Proceedings of the ROCLING 2011 Poster Papers*. Association for Computational Linguistics, 252–264.
[23] Kathleen A. Hansen, Sarah F. Hillenbrand, and Leslie G. Ungerleider. 2012. Effects of prior knowledge on decisions made under perceptual vs. categorical uncertainty. *Frontiers in Neuroscience* 6 (2012), 163.
[24] Moritz Hardt, Eric Price, and Nathan Srebro. 2016. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*. 3315–3323.
[25] Florian Heimerl, Steffen Koch, Harald Bosch, and Thomas Ertl. 2012. Visual classifier training for text document retrieval. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2839–2848.
[26] Jessica Hullman. 2013. How prior knowledge affects the processing of visualized data. In *Proceedings of the ACM CHI 2013, Many People Many Eyes Workshop*.
[27] Dong Hyun Jeong, Caroline Ziemkiewicz, Brian Fisher, William Ribarsky, and Remco Chang. 2009. iPCA: An Interactive System for PCA-based Visual Analytics. In *Computer Graphics Forum*, Vol. 28. Wiley Online Library, 767–774.
[28] John T. Kent. 1983. Information gain and a general measure of correlation. *Biometrika* 70, 1 (1983), 163–173.
[29] Sotiris B. Kotsiantis, I. Zaharakis, and P. Pintelas. 2007. Supervised machine learning: A review of classification techniques. In *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*.

[30] Josua Krause, Adam Perer, and Enrico Bertini. 2014. INFUSE: Interactive feature selection for predictive modeling of high dimensional data. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 1614–1623.

[31] Josua Krause, Adam Perer, and Kenney Ng. 2016. Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 5686–5697.

[32] 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*. ACM, 3075–3084.

[33] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. 2015. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*. ACM, 126–137.

[34] Todd Kulesza, Simone Stumpf, Weng-Keen Wong, Margaret M Burnett, Stephen Perona, Andrew Ko, and Ian Oberst. 2011. Why-oriented end-user debugging of naive Bayes text classification. *ACM Transactions on Interactive Intelligent Systems* 1, 1 (2011), 2.

[35] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Eric Horvitz. 2017. Identifying unknown unknowns in the open world: Representations and policies for guided exploration. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, Vol. 1. 2124–2132.

[36] Rensis Likert. 1932. A technique for the measurement of attitudes. In *Archives of Psychology*. The Science Press, New York.

[37] Brian Y. Lim, Anind K. Dey, and Daniel Avrahami. 2009. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2119–2128.

[38] Shixia Liu, Xiting Wang, Mengchen Liu, and Jun Zhu. 2017. Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics* 1, 1 (2017), 48–56.

[39] 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-Volume 1*. Association for Computational Linguistics, 142–150.

[40] Frank J. Massey Jr. 1951. The Kolmogorov–Smirnov test for goodness of fit. *Journal of the American Statistical Association* 46, 253 (1951), 68–78.

[41] Thorsten May, Andreas Bannach, James Davey, Tobias Ruppert, and Jörn Kohlhammer. 2011. Guiding feature subset selection with an interactive visualization. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST'11)*. IEEE, 111–120.

[42] Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. 2006. Spam filtering with naive Bayes—which naive Bayes? In *Proceedings of the 3rd Conference on Email and Anti-Spam (CEAS'06)*, Vol. 17. 28–69.

[43] Mohammad Sadegh, Roliana Ibrahim, and Zulaiha Ali Othman. 2012. Opinion mining and sentiment analysis: A survey. *International Journal of Computers and Technology* 2, 3 (2012), 171–178.

[44] Ted O'Donoghue and Matthew Rabin. 1999. Doing it now or later. *American Economic Review* 89, 1 (1999), 103–124.

[45] Kayur Patel, James Fogarty, James A. Landay, and Beverly Harrison. 2008. Investigating statistical machine learning as a tool for software development. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 667–676.

[46] 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.

[47] Brett Poulin, Roman Eisner, Duane Szafron, Paul Lu, Russell Greiner, David S. Wishart, Alona Fyshe, Brandon Pearcy, Cam MacDonell, and John Anvik. 2006. Visual explanation of evidence with additive classifiers. In *Proceedings of the National Conference on Artificial Intelligence*, Vol. 21. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press.

[48] Hema Raghavan and James Allan. 2007. An interactive algorithm for asking and incorporating feature feedback into support vector machines. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 79–86.

[49] Hema Raghavan, Omid Madani, and Rosie Jones. 2005. InterActive feature selection. In *Proceedings of the 19th international joint conference on Artificial intelligence (IJCAI'05)*, Vol. 5. 841–846.

[50] Hema Raghavan, Omid Madani, and Rosie Jones. 2006. Active learning with feedback on features and instances. *Journal of Machine Learning Research* 7, Aug. (2006), 1655–1686.

[51] Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment* 11, 3 (2017), 269–282.

[52] Donghao Ren, Saleema Amershi, Bongshin Lee, Jina Suh, and Jason D. Williams. 2017. Squares: Supporting interactive performance analysis for multiclass classifiers. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 61–70.

[53] Juha Reunanen. 2003. Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research* 3, Mar. (2003), 1371–1382.
[54] Xin Rong and Eytan Adar. 2016. Visual tools for debugging neural language models. In *Proceedings of ICML Workshop on Visualization for Deep Learning*.
[55] Burr Settles. 2011. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1467–1478.
[56] Eldar Shafir. 2007. Decisions constructed locally: Some fundamental principles of the psychology of decision making. In *Social Psychology: Handbook of Basic Principles*. Arie W. Kruglanski and E. Tory Higgins (Eds.), Guilford Publications, Chapter 14, 334–352.
[57] Patrice Y. Simard, David Maxwell Chickering, Aparna Lakshmiratan, Denis Xavier Charles, Léon Bottou, Carlos Garcia Jurado Suarez, David Grangier, Saleema Amershi, Johan Verwey, and Jina Suh. 2014. ICE: Enabling non-experts to build models interactively for large-scale lopsided problems. CoRR abs/1409.4814.
[58] Herbert A. Simon. 1957. *Models of Man; Social and Rational*. Wiley, Oxford.
[59] Simone Stumpf, Adrian Bussone, and Dympna O 'sullivan. 2016. Explanations considered harmful? User interactions with machine learning systems. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'16)*.
[60] Simone Stumpf, Vidya Rajaram, Lida Li, Weng-Keen Wong, Margaret Burnett, Thomas Dietterich, Erin Sullivan, and Jonathan Herlocker. 2009. Interacting meaningfully with machine learning systems: Three experiments. *International Journal of Human-Computer Studies* 67, 8 (2009), 639–662.
[61] Simone Stumpf, Erin Sullivan, Erin Fitzhenry, Ian Oberst, Weng-Keen Wong, and Margaret Burnett. 2008. Integrating rich user feedback into intelligent user interfaces. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*. ACM, 50–59.
[62] Google Big Picture Team. 2017. Facets: Visualization for ML datasets. Retrieved February 11, 2018 from https://pair-code.github.io/facets/.
[63] Gaurav Trivedi, Phuong Pham, Wendy Chapman, Rebecca Hwa, Janyce Wiebe, and Harry Hochheiser. 2015. An interactive tool for natural language processing on clinical text. In *Proceedings of the 4th Workshop on Visual Text Analytics (IUI TextVis'15)*. Retrieved from http://vialab.science.uoit.ca/textvis2015/[PDF].
[64] Joe Tullio, Anind K. Dey, Jason Chalecki, and James Fogarty. 2007. How it works: A field study of non-technical users interacting with an intelligent system. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 31–40.
[65] Amos Tversky and Daniel Kahneman. 1974. Judgment under uncertainty: Heuristics and biases. *Science* 185, 4157 (1974), 1124–1131.
[66] Paroma Varma, Dan Iter, Christopher De Sa, and Christopher Ré. 2017. Flipper: A systematic approach to debugging training sets. In *Proceedings of the 2nd Workshop on Human-in-the-Loop Data Analytics*. ACM, 5.
[67] Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin* 1, 6 (1945), 80–83.

Received March 2018; revised October 2018; accepted March 2019