Towards Human-Guided Machine Learning

Yolanda Gil[1], James Honaker[2], Shikhar Gupta[1], Yibo Ma[1], Vito D'Orazio[3], Daniel Garijo[1], Shruti Gadewar[1], Qifan Yang[1], Neda Jahanshad[1]

[1]University of Southern California
Los Angeles, California, USA
{gil, dgarijo}@isi.edu
{shikharg, yiboma, gadewar, qifan.yang, neda.jahanshad}@usc.edu

[2]Harvard University
Boston, Massachusetts, USA
jhonaker@iq.harvard.edu

[3]University of Texas at Dallas
Dallas, Texas, USA
dorazio@utdallas.edu

ABSTRACT

Automated Machine Learning (AutoML) systems are emerging that automatically search for possible solutions from a large space of possible kinds of models. Although fully automated machine learning is appropriate for many applications, users often have knowledge that supplements and constraints the available data and solutions. This paper proposes human-guided machine learning (HGML) as a hybrid approach where a user interacts with an AutoML system and tasks it to explore different problem settings that reflect the user's knowledge about the data available. We present: 1) a task analysis of HGML that shows the tasks that a user would want to carry out, 2) a characterization of two scientific publications, one in neuroscience and one in political science, in terms of how the authors would search for solutions using an AutoML system, 3) requirements for HGML based on those characterizations, and 4) an assessment of existing AutoML systems in terms of those requirements.

CCS CONCEPTS

• **Human-centered computing**

KEYWORDS

Human-guided machine learning; Automated machine learning (AutoML); Task analysis; Scientific workflows.

ACM Reference format:

Yolanda Gil, James Honaker, Shikhar Gupta, Yibo Ma, Vito D'Orazio, Daniel Garijo, Shruti Gadewar, Qifan Yang, Neda Jahanshad. 2019. Towards Human-Guided Machine Learning. In *24th International Conference on Intelligent User Interfaces (IUI '19), March 17–20, 2019, Marina del Rey, CA, USA. ACM, New York, NY, USA, 11 pages.* https://doi.org/10.1145/3301275.3302324

IUI '19, March 17–20, 2019, Marina del Rey, CA, USA

ACM ISBN 978-1-4503-6272-6/19/03.
https://doi.org/10.1145/3301275.3302324.

1 Introduction

In recent years, Automated Machine Learning (AutoML) approaches are making great strides to automatically search for machine learning solutions from a large space of possible kinds of models. Typically, a solution is created by choosing a model (e.g., random forest, SVM, etc.) and then configuring those models by assigning (hyper)parameter values [1]–[4].

A series of challenges and workshops have led to steadfast improvements [5]. Commercial products are now becoming available that automate machine learning, notably for image classification and select natural language processing tasks [6]. While fully automated model learning is appropriate for many applications, there are many contexts where full automation is not desirable or possible.

This is the case when users have knowledge that supplements the available data, particularly in a scientific research context. An AutoML system would look at a set of instances or images the same whether they are about tumor tissues or ad placements. However, biologists would bring to bear extensive knowledge about human disease in the development of a machine learning model. Without this knowledge, machine learning models might optimize model search criteria but be inconsistent with what is known about the data and its context. Moreover, without incorporating this knowledge the solution search space might be intractable. In many cases, users explore alternative settings (different models, different features and different datasets) in order to consider a variety of questions and hypotheses.

This paper proposes human-guided machine learning (HGML) as a hybrid approach where a user interacts with an AutoML system and tasks it to explore different problem settings that reflect the user's knowledge about the data available. This requires an intelligent user interface that allows users to specify alternative problem settings and explore different models, and an AutoML system that can be tasked to generate solutions according to the user's guidance. We explore requirements for HGML based on two substantial scientific publications in different disciplines (political science and neuroscience), and analyze how those requirements could be met by AutoML systems.

The main contributions of this paper are: 1) An integrated user interface and AutoML system that supports basic interactions

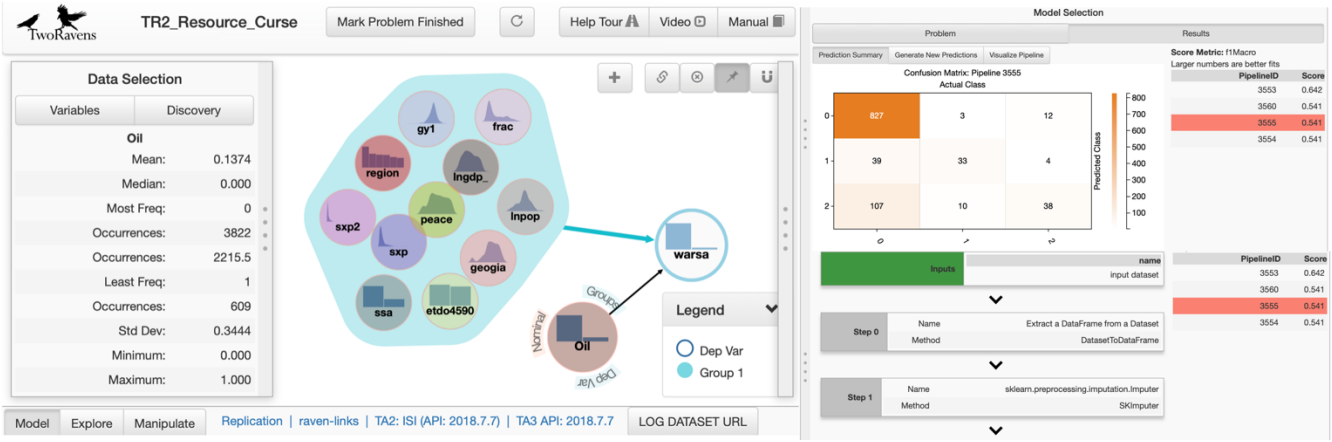

Figure 1: **Our initial integrated system that serves as a baseline for Human-Guided Machine Learning (HGML).**

as a baseline for HGML; 2) A task analysis of HGML that enumerates discrete user tasks to guide AutoML systems; 3) Characterizations of two significant studies in neuroscience and political science based on that task analysis; 4) Requirements for HGML resulting from those characterizations; and 5) An assessment of how those requirements could be accommodated by AutoML systems.

The paper begins with a description of an existing AutoML system that supports basic interactions with users. We then motivate why users could use their knowledge to effectively guide the search for machine learning models. We follow with a task analysis of common tasks involved in HGML, mapping these tasks to the steps followed in two recent publications in different disciplines. We then assess whether existing AutoML systems could accommodate those requirements. We finish with a discussion and an outline of future work.

2 Automating Machine Learning

A variety of approaches to AutoML have been explored in recent years. Auto-WEKA [1] combines automated model selection with (hyper)parameter optimization using the machine learning functions or primitives provided by the popular WEKA toolkit [7]. The approach was adopted for the scikit-learn library [8] in Auto-sklearn [2], extended with meta-learning and ensemble construction, and later with probabilistic matrix factorization [4]. TPOT offers an alternative approach using genetic algorithms [3].

3 Interacting with an AutoML System

In prior work, we integrated an AutoML system [10] with a user interface for statistical and machine learning model exploration [9] to allow users to specify machine learning problems of interest and have the AutoML system generate solutions. Figure 1 shows a snapshot of this system. The left-hand side summarizes the variables in a dataset, the middle portion highlights statistical properties of each variable (which can be explored further by the user), and the right-hand side shows how a user can specify a machine learning problem of interest. With that specification, the AutoML system returns the top ranked multi-step solutions that include an appropriate machine learning model.

Our AutoML system uses a phased approach to AutoML that can ingest naturally occurring data of any size and type, and use third-party data processing and modeling primitives [10]. In contrast with previous approaches to AutoML, this system can ingest different kinds of data such as tables, text, images and audio. The system exploits expert-like planning strategies to factor the search for multi-step solutions into phases: 1) extract features of interest from the dataset, 2) build a solution skeleton with the types of model and other steps to include, 3) address algorithm requirements with additional steps, 4) perform a hyperparameter search to improve the results, and 5) generate ensembles with the top-ranked models. The planner generates multi-step solutions that include featurization steps, data cleaning steps, imputation steps and modeling steps.

In this integrated system, a user specifies: 1) A target dataset with many instances (or data points); 2) A problem description (currently this can be classification or regression); 3) A specification of training and test data as subsets of the target dataset, including cross-validation details (number of folds, stratified or plain); and 4) An evaluation metric, which can be accuracy, mean squared error, and F1 macro. The system returns the top ranked solutions. Below are some example solutions for a classification problem to map questions to answers, showing the main steps and accuracy:

HashingVectorizer -> LabelEncoder -> LogisticRegressionCV (0.9489)

CountVectorizer -> LabelEncoder -> BernoulliNB (0.9486)

TfidfVectorizer -> LabelEncoder -> AdaBoostClassifier (0.9460)

This integrated system provides a baseline for HGML, in that it allows users to task the AutoML planner to find a machine learning solution but does not support other kinds of guidance from users. In the next section we motivate why this guidance is key in many applications.

4 Human-Guided Machine Learning

We view human-guided machine learning (HGML) as a new area of research focused on how to assist users to use domain knowledge to guide an AutoML system to select machine learning algorithms and find multi-step solutions. Exploiting domain knowledge is particularly important for scientific data, which have been our focus to date.

There is important domain knowledge that can inform a predictive model but cannot be derived directly from a dataset or from abstract statistical or learning theory. This includes substantive theories about the context of the data, past results in the literature that lend prior weight to different factors and models, the purpose and properties of a good model, and the applicability of solutions from an AutoML system. This knowledge is complementary to the data itself.

In today's very best research settings, where there are collaborations between domain experts and machine learning experts, exploration into the data is a joint venture where machine learning experts drive the computational machinery of analysis, but are directed to the interesting features by the knowledge of the domain experts. Often domain experts do not even realize what knowledge is most valuable, or what its implications are for tailoring the statistical choices, but a good machine learning collaborator can untangle and extract that information through discussion and model exploration. The goal of our work is to increasingly automate what a machine learning expert contributes to this process, so that a domain expert can develop machine learning models on their own.

Domain experts are typically heavily invested in the collection of their data, understand the variables well, pose questions that could be answered from the data, and possess substantive knowledge about the plausibility of various relationships that may emerge from models. What they may not know is the range of plausible models appropriate to their objectives, how to configure and parameterize models to explore the solution space, or how to interpret their results. We could imagine a hypothetical consulting machine learning expert whose role is to tease out all of the expert knowledge the researcher possesses, address any data quality issues, choose appropriate machine learning models, and then conduct tests as to whether the model has any failings. Alas, the extreme overabundance of data in many domains greatly surpasses the availability of such individuals, leaving many data-rich questions unresolved.

Our goal is to develop HGML systems that allow a domain expert, without a machine learning expert, to use relevant domain knowledge to inform the automated search for a high quality, impactful and interpretable model, including necessary data preparation steps necessary for analysis.

An HGML approach assumes iterative exploration and incremental development of models. That is, it does not assume a one-shot problem set up by the user and solution generation by the system. Users are expected to explore a variety of settings and models in order to understand the data and to formulate appropriate questions and hypotheses.

5 Related Work

The most closely related work is on interactive machine learning research and human-in-the-loop machine learning, a thorough review is provided in [11]. A variety of work has focused on interactive clustering, where users interact with a single type of clustering model, typically by requesting that clusters be split or merged. [16] focus on assisting a user to select parameter value ranges for clustering based on visualizations of a variety of metrics. In the case of clustering text data, there are approaches that allow users to provide additional kinds of input, such as term-based feedback on topic models [17]–[18]. Other work has focused on interactive feature selection for machine learning by developing user models based on prior interactions [19]. These approaches focus on helping a user set up and interact with machine learning models, while HGML allows users to task an AutoML system to do so.

Visual analytics focuses on approaches to generate visualizations that can help a user understand and analyze data, and even transform the data through manipulation [12]–[15]. Others have focused on exploring data through queries and statistic tests [20]–[21]. This exploration can lead a user to define new classes or features [22]. This work is complementary to HGML in that it focuses on helping users decide what features or classes to use rather on enabling them to specify features based on their knowledge.

There is also work on learning procedures from user logs or from demonstration [23]–[25], which could be the basis for HGML approaches that learn from watching users create end-to-end machine learning solutions in their domains.

In summary, prior work addresses issues of users providing input or reacting to a model generated by a machine learning algorithm, using the terms "interactive machine learning" or "human-in-the-loop machine learning". There is no interaction on data preparation steps, which are known to affect the performance of machine learning models. There are also no choices of types models, as users interact with a single model (e.g., a k-means clustering model). In contrast, human-guided machine learning focuses on the user providing guidance to a fully automated machine learning system. Our work shifts the attention to maximizing machine learning automation, and takes the user interactions further to collaborate with an AutoML system that can: 1) take on a variety of machine learning tasks on a wide range of types of data, 2) explore several model types, and 3) consider entire end-to-end solutions that include data preparation and feature generation steps.

6 Task Analysis

This section describes our task analysis work to characterize common tasks involved in human-guided machine learning. We followed two phases, the first was top-down and the second bottom up.

Table 1 summarizes the resulting tasks, showing in boldface the tasks that resulted from the second phase of the analysis.

Table 1. **Major Tasks Involved in Human-Guided Machine Learning (HGML).**

Task Category & Purpose		Task ID and description	User reason	Intended Guidance
Data Use	Variable Selection	[VS1] The user may require that certain variables be given more priority	User's domain knowledge indicates the importance of specific variables	Affect feature selection and weighting
		[VS2] The user may suggest new variables that combine existing variables	User's domain knowledge and experience indicates new variables	Affect feature generation
		[VS3] The user may modify the framing (e.g., temporal, spatial) in which the variables exist	User is interested in a particular time period or area	Affect feature selection
		[VS4] The user may fill in missing values in data based on existing variables	User's domain knowledge indicates other variables may be good indicators	Affect missing value imputation
		[VS5] The user may augment the available data with new variables	User may know of related data sources that can be brought to bear	Affect data and feature availability
	Variable Requirements	[VR1] The user may ask that certain variables be included or excluded in a model	User's domain knowledge indicates certain variables are good/bad predictors for the task at hand	Affect feature selection/filtering
	Instance Selection	[IS1] The user may highlight instances that need to be accounted for in the model	The user is interested in target instances for the analysis at hand	Affect the instances used to train a model
		[IS2] The user may select a subpopulation based on properties of the instances	User's knowledge indicates sub-population may lead to more precise analysis	Affect the instances used to train a model
		[IS3] The user may remove specific instances (e.g., outliers, poor quality, etc.)	The user believes that certain instances in the dataset will bias the model in undesirable ways	Affect the instances used to train a model
		[IS4] The user selects training and test data, optionally with cross-validation specifications	The user may want to explore different training and test data splits of the data	Affect the instances used to train and test a model
Model Development	Model Selection	[MS1] The user may request a specific type of model	Users may have preferences (e.g., for a causal model or for an explanatory model)	Affect the models included in the solution
		[MS2] The user may request a specific type of model to replace the model in a prior solution	User question of interest may redefine what models are appropriate to the task at hand	Affect the models included in a solution
	Parameter Settings	[PT1] The user may request the use of specific parameter values	User's knowledge may indicate target parameters to reach a good solution efficiently	Configure models included in the solution
	Solution Design	[SD1] The user may request specific data preparation steps (primitives or more general steps) to be present in the solution	User's domain knowledge indicates that a certain feature generation or imputation step may be required	Affect the models included in the final solution
		[SD2] The user may request to replace a data preparation step in a prior solution with a new step	User domain knowledge may assist search, or user may want to investigate consequences of model choices	Affect the models included in the final solution
Model Interpretation	Model Assessment	[QC1] The user may request quantities of interest to be computed	Users may request metrics, the regression slope, or ask for other means of interpreting a model	Perform computations on the data or model
		[QC2] The user may request certain intermediate results that may be of interest	Intermediate results may indicate how a solution is analyzing data, and what features are being considered	Record intermediate results of the analysis
	Model Comparison	[MC1] The user may ask for two or more solutions each with a different type of model	Users may want to compare alternatives (e.g., based on model performance or explainability)	Generate several solutions
		[MC2] The user may ask for two or more solutions with a given model but each trained on different subsets of the instances	Users may want to compare how models differ when trained with different data	Generate several solutions
		[MC3] The user may ask to compare two or more solutions with the same model but different data preparation steps	Users may want to find an explanation for the results provided to understand the different steps that appear in different solutions	Compare provenance records (i.e., steps, parameters, etc.) of two solutions
		[MC4] The user may ask to compare two or more solutions with the same model but different parameters	User wants to understand the sensitivity of the model to different parameters	Compare provenance records of solutions
		[MC5] The user may ask for a comparison between two models, all other solution steps being the same	Users may want an explanation of why two models have different performance	Generate a comparison of models
	Parameter Comparison	[PC1] The user may ask for several solutions, each with the same model or model primitive but a range of parameter values for one or more parameters	Users may want to validate a solution by verifying that is not significantly affected by a given parameter	Generate several solutions

We omit for brevity 2 tasks concerning hypothesis generation and testing, since they are not yet supported by existing AutoML systems. A detailed description of the tasks is available in [26].

6.1 First Phase: Initial Task Analysis

In the first phase, we took a top-down approach and articulated and categorized an initial set of tasks based on our understanding of how users work with machine learning algorithms to develop applications and our expectation of how they would want to interact with an AutoML framework. Table 1 shows the resulting 13 tasks (in light typestyle), grouped under general purpose and three major categories:

1. *Data Use:* Tasks concerning use of data allow users to specify which observations (rows) and features (columns) should be used to generate a machine learning model specific to the user's foundational question of interest. For example, experts might need to remove data points that are not relevant to the question at hand, or give priority to observations that are more important to correctly predict. They might wish to include certain variables based on prior knowledge in the literature or to explicitly test a theory of interest, or remove variables they know cannot be related to increase the speed of the automated model search [27].
2. *Model Development*: Model development tasks broadly involve the user constraining the space of possible models beyond what could automatically be computed. Users may have prior knowledge or judgements they can provide that (directly or indirectly) help to tune model parameters. More importantly, the underlying purpose of their analysis, (which only they know) may change what models are appropriate. For example, in causal analysis there is a need for machine learning models that avoid bias [28].
3. *Model Interpretation*: Interpretation of machine learned models allows the user to guide how diagnostics and predictions from the model best illuminate the questions of interest the expert is interested in understanding. Also, directly comparing the performance of differently constructed models allows the researcher to better understand what the models infer about the world.

6.2 Second Phase: Annotation of Research Publications

In this phase, we followed a bottom-up approach and analyzed two very different publications where the authors describe a series of models, and annotated the author's activities according to our tasks based on the author's notes in the papers. That is, we tried to envision what the interaction would be between the article authors and an HGML system. This used the 13 tasks we had categorized, and added 10 more tasks shown in boldface in Table 1. A more detailed analysis is provided in [29].

6.2.1 Neuroscience Analysis. learning analysis produced by the ENIGMA neuroscience consortium [32]. ENIGMA aims to conduct large-scale collaborative research studies that show reliable and reproducible findings with greatly improved statistical power. The article analyzed here [30]-[31] aims to come to an unbiased consensus on the extent to which brain scans can classify patients with major depressive disorder (MDD) in the largest study of its kind.

Data for the consortium is pooled from independent MDD studies from around the world, allowing for a combined dataset of over 2,500 samples. Each individual study, or "cohort", recruited both healthy individuals and patients with clinically diagnosed depression, and collected MRI and clinical data according to independent criteria (e.g., some were studies of younger adults, some older, some had no age restriction, some were more medicated, some were unmediated, etc.; all were studies of MDD). While many classification attempts have shown over 80% accuracy in MDD classification from imaging measures (see [33] for an extensive review), most studies had been limited in power with less than 50 samples in each group and lacked independent replication in a separate sample. Furthermore, publication bias where null or unremarkable results are not published, has limited reports of more modest or unsuccessful classification attempts.

Table 2 gives an overview of our analysis of the article, showing the identifier for the user interaction, the task type corresponding to the tasks in Table 1, and a brief description. The authors explore 6 different models.

The authors choose one cohort in the list of cohorts as the test data and all the other cohorts as the training data. This is done to prevent data leakage, such that the test data was collected on a brain scanner that is different from those used in the training data, and had its own specifications on patient and control inclusion and exclusion criteria. In other words, the classification is performed to evaluate MDD diagnosis independent of specific study conditions. Leaving out an entire cohort for testing could reflect the actual performance of the models on unseen data collection.

After conducting the classification using the entire set of samples, the authors further refined their models to be trained and tested, or just tested, on a specific subset of samples determined by clinical, non-imaging related features. In Model 3, 4, 5, and 6 the classification was set according to factors such as the sex of the participants and whether the MDD patients had multiple depressive episodes. By comparing their performance, the generalizability and specificity of the models is exposed.

The authors did not include non-imaging features (i.e., age, sex) in models 1 and 2, but those features were included in models 3, 4, 5 and 6. While the imaging features were of primary interest, factors such as sex and age are related to the distribution of the imaging brain measures, so by comparing models 1 and 2 versus models 3, 4, 5, and 6, the authors could compare the performance of imaging measures alone to models that also account for basic patient demographic information, which would almost always be available in a clinical setting.

6.2.2 Political Science Analysis. In [34] Fearon provides a quintessential style of analysis in social science, which we call "empirical robustness." The author is interested in whether a hypothesis is supported across alternative data and modeling specifications. The hypothesis explored is that primary

Table 2. **Task Analysis for Neuroscience Article.**

ID	Task Type	Description
Model 1 (Lasso + SVM with brain image features)		
N1	VR1	Leave out non-brain-image columns ("sex" and "Episodes").
N2	IS2/VR1	Eliminate subjects whose age is over 21, and then drop the column "age".
N3	IS3	Drop all rows with missing values.
N4	IS2	Set up training set as all cohorts (sites) but one, each time leaving out a cohort as test set.
N5	VR1	Drop columns "subjid" and "site" from both training data and test data.
N6	IS4	Use stratified cross validation to evaluate the model trained on partial training set.
N7	MS1	Run Lasso on training set to gain feature importance (weights).
N8	VS1	Use the feature weights produced by Lasso and larger than 0 to select important features for modeling. If there are more than 5 features whose weights are large than 0, only keep the most important 5 features.
N9	MS1/PT1	Build a SVM model with specified parameters: kernel=rbf, gamma={1e-1,1e-5}, c={1,10,100,1000} for a grid search.
N10	QC1	Use multiple metrics (sensitivity, specificity, accuracy, balanced accuracy, F1 score, Matthews correlation coefficient and confusion matrix) to evaluate model performance. Accuracy, F1 score and Confusion Matrix are used to evaluate the model directly and other metrics are for comparison to previous publication.
N11	PD1	Perform N7 to N10 on the whole training set and use the test set to evaluate model performance.
Model 2 (Logistic Regression with brain image features)		
N1 – N6 same as above		
N12	MS1/PT1	Build Logistic Regression model with specified parameters, penalty=l1, solver=liblinear, class_weight=balanced.
N10 same as above		
N13	PD1	Perform N12 and N10 on the whole training set and use the test set to evaluate model performance.
Model 3 (Lasso + SVM with brain image features and splitting test set based on recurrence)		
N14	VR1	Leave out the non-brain-image column "sex".
N15	VS2/VR1	Generate variable "recur" based on the variable "Episodes". "Episodes" represents the number of times a disease occurred. If the disease occurred more than once, the "recur" value is 2. Otherwise, the "recur" value is 1, meaning the disease just occurred once on the subject. Then drop the variable "Episodes".
N2 – N10 same as above		
N16	IS2	Split all test instances into two groups by their "recur" values (1 and 2). Then the user can observe if there is a difference in predictions between the two groups of test data, which can show the influence of the "recur" feature to the model.
N11 same as above		
Model 4 (Logistic Regression with brain image features and splitting test set based on recurrence)		
N14 - N15 and N2 – N6 and N12 and N10 and N16 and N13 same as above		
Model 5 (Lasso + SVM with brain image features and splitting test set based on sex)		
N17	VR1	Leave out the non-brain-image column "Episodes".
N2 – N10 same as above		
N18	IS2	Split all test instances into two groups by their "sex" values (1 or 2). Then the user can observe if there is a difference in predictions between the two groups of test data, which can show the influence of the "sex" feature to the model.
N11same as above		
Model 6 (Logistic Regression with brain image features and splitting test set based on sex)		
N17 and N2 – N6 and N12 and N10 and N18 and N13 same as above		

commodity exports (sxp) increase the likelihood of civil waronset. Earlier work by Collier and Hoeffler [35] argue sxp to be *the primary* driver of civil war risk. To demonstrate the fragility of their findings, Fearon first identifies points where Collier and Hoeffler chose one of several viable paths in model construction. For example, Collier and Hoeffler use a five-year data aggregation when a single-year aggregation is equally justifiable. He then shows that the empirical relationship between sxp and civil war onset is not robust under these alternative specifications.

While AutoML systems could take Fearon's or Collier and Hoeffler's data and produce a model to predict the onset of civil conflict, they would not be able to identify the policy relevance of sxp, the widespread impact of Collier and Hoeffler's work, and thus the need to thoroughly assess the finding's fragility. Furthermore, Fearon's expertise with this type of data and his substantive knowledge of civil wars allowed him to identify a meaningful set of alternative viable paths. The domain knowledge that he brings to bear on this problem is outside, or auxiliary to, the data itself.

Table 3 shows the interaction tasks for each of the steps involved in creating all the models reported in Fearon's article. He begins with a replication of Collier and Hoeffler's base model (Step F1). Here, sxp and sxp^2 are each statistically significant, and a likelihood ratio test shows the pair is jointly significant. The same holds for a model containing only sxp and sxp^2 as predictors (F2). He adds $\log(population)$ to this reduced model (F3) and shows comparably sized effects for sxp and sxp^2 between this reduced model and the base model. He splits the data into treciles by population and sxp/GDP to show that the core empirical finding is present only for large states. Then, Fearon explores a number of alternative viable modeling decisions to show the fragility of this base finding.

Briefly, he shows that $\log(sxp)$ is statistically significant (F6) and uses a generalized additive model to provide some empirical support for using $\log(sxp)$instead of sxp and sxp^2 (F5). He disaggregates the data from five-year to one-year observations (F7) and re-estimates the baseline model (F8), along with

Table 3. **Task Analysis for Political Science Article.**

ID	Task Type	Dataset	Description of Interaction
TABLE 2 MODEL 1 (Collier and Hoeffler Model Replication)			
F1	VR1, MS1, QC1	DS1	Fearon develops a logistic regression model with specified variables from Collier and Hoeffler's dataset (**DS1**) and tests the joint significance of primary commodity exports (sxp) and sxp^2 with a likelihood ratio test (χ^2 test).
F2	VR1, MS1, QC1	DS1	Fearon determines F1 feature significance by developing a logistic regression model using sxp and sxp^2 and runs a χ^2 test. He notices a weak relationship, but unlikely due to chance.
F3	VR1, MS1, QC1	DS1	Fearon adds $\log(population)$ to the F2 model, resulting in sxp's coefficient approaching the value in F1. He runs a χ^2 test, noticing the impact of sxp and $\log(population)$, with civil war onset probability at .11 in the 90th percentile of sxp.
F4	IS2	DS1	Fearon hypothesizes χ^2 test in F3 passed due to the impact of large countries, as they have low commodity exports as a percentage of GDP, but a higher civil war risk. He creates TABLE 3, which includes the proportion of civil war onset at the intersections of country population and sxp treciles.
TABLE 2 MODEL 2 (Adjusting from Parabolic to log Relationship)			
F5	MS1, PT1	DS1	Collier and Hoeffler note a parabolic relationship between sxp and $\log(odds)$ of civil war onset. Fearon proves them incorrect, instead hypothesizing an inverse log relationship and proving so using a generalized additive model (GAM).
F6	VS2, MS1, QC1	DS1	Fearon concludes $\log(sxp)$ should be created to replace sxp and sxp^2, creating dataset **DS2**, and used with the other dependent variables in a logistic regression model. Fearon then runs a χ^2 test, noticing a reduced correlation.
TABLE 2 MODEL 3 (Making Temporal Framing Adjustments)			
F7	VS3, PD1	DS1	Fearon adjusts temporal framing of data to yearly periods, as five-year periods cause issues with quickly renewed wars and expand variable lag times. Many variables are measured annually or are time invariant and 75% of sxp's variation is across countries. With $r = .85$ for sxp in year t vs. $t - 5$, he uses linear interpolation or spline to fill missing values, creating **DS3**.
F8	VR1, MS1, QC1	DS3	Fearon includes specified variables in the logistic regression model. The coefficient for fractionalization falls by a factor of three, while for geographic dispersion is close to zero. Fearon runs a χ^2 test, failing to reject the null hypothesis.
TABLE 2 MODEL 4, 5, and 6 (Dropping Control Variables)			
F9	VR1, MS1, QC1	DS3	Fearon drops fractionalization, ethnic dominance, and geographic dispersion from F8 and runs a χ^2 test.
TABLE 4 MODEL 1, 2, and 3 (Dropping Control Variables with an adjusted log Relationship)			
F10	VS2, MS1, QC1	DS3	Fearon adjusts TABLE 2 MODEL 3-6 by replacing sxp and sxp^2 with $\log(sxp)$, creating **DS4**, and creates a logistic regression model with other dependent variables. He then runs a χ^2 test.
F11	VR1, MS1, QC1	DS4	Fearon drops fractionalization, ethnic dominance, and geographic dispersion from the logistic regression model. He runs a χ^2 test, noting lower statistical significance than TABLE 2 MODEL 6.
TABLE 5 MODEL 1 (Filling Missing Data with Multiple Imputation)			
F12	PD1	DS1	He uses multiple imputation to highlight the "significance" of sxp based on list-wise deletion of observations, creating **DS5**.
F13	VR1, MS1, QC1	DS5	Fearon creates a logistic regression model with specified variables and runs a χ^2 test. Coefficients move towards zero, sxp and sxp^2 drop by a factor of two, and the 10th to 90th percentile changes from 1.1 to 11 percent to 2.3 to 7.4 percent.
TABLE 5 MODEL 2 (Dropping Control Variables on Modified Dataset)			
F14	VR1, MS1, QC1	DS5	Fearon drops fractionalization, ethnic dominance, and geographic dispersion from F13 and runs a χ^2 test.
TABLE 5 MODEL 3 (Adjusting Temporal Frame on Modified Dataset)			
F15	VS3, PD1	DS5	Fearon adjusts the temporal framing to single years and uses multiple imputation or spline for the missing data, creating **DS6.**
F16	VR1, MS1, QC1	DS6	Fearon creates a logistic regression model with specified variables and runs a χ^2 test, noting a lack of statistical significance.
TABLE 5 MODEL 4 (Dropping Control Variables from Temporally Modified Dataset)			
F17	VR1, MS1, QC1	DS6	Fearon drops fractionalization, ethnic dominance, and geographic dispersion from F16 and runs a χ^2 test.
TABLE 6 MODEL 1 (Isolating Oil Exports from sxp)			
F18	VS2, MS1, QC1	DS1	Fearon creates a variable calculated as a percentage of fuel exports vs. total exports, creating **DS7**, because high oil dependence is associated with higher civil war risk. He runs a logistic regression with dependent variables and a χ^2 test.
TABLE 6 MODEL 2 (Dropping Control Variables from Isolated Dataset)			
F19	VR1, MS1, QC1	DS7	Fearon drops fractionalization, ethnic dominance, and geographic dispersion from F18 and runs a χ^2 test.
TABLE 6 MODEL 3 (Adjusting Temporal Frame on Isolated Dataset)			
F20	VS3, PD1	DS7	He adjusts the temporal framing to single years and uses multiple imputation or spline to fill the missing data, creating **DS8.**
F21	VR1, MS1, QC1	DS8	Fearon creates a logistic regression model with specified variables and runs a χ^2 test, with oil exports and sxp trading places.
TABLE 6 MODEL 4 (Adding log Relationship to Isolated Dataset)			
F22	VS2, MS1, QC1	DS8	Fearon creates $\log(sxp)$, creating **DS9**, creates a logistic regression model with dependent variables, and runs a χ^2 test.
TABLE 6 MODEL 5 (Dropping Control Variables from Isolated Dataset)			
F23	VR1, MS1, QC1	DS9	Fearon drops fractionalization, ethnic dominance, and geographic dispersion from F22 and runs a χ^2 test. While sxp is insignificant, there is strong support for a nonrandom association between high oil exports and civil war risk.
TABLE 7 MODEL 1 (Hypothesis Testing Government Observance of Contracts)			
F24	VR1	DS2	Fearon disputes Collier and Hoeffler's argument on rebel financing through cash crops/fuel exports, proving sxp dependence for national income marks weak state institutions by using a measure of a state's administrative capability and integrity.
F25	MS1	DS2	Fearon creates a regression model for the measure of contract observance on $\log(sxp)$ and $\log(income)$.
TABLE 7 MODEL 2 (Adding Fuel Exports to Government Observance of Contracts)			
F26	VR1	DS2	Fearon adds fuel exports as another measure of weakness to F25.
F27	MS1	DS2	Fearon creates a regression model, with the estimated coefficient for sxp becoming statistically insignificant, but fuel exports maintaining significance. Oil exporters thus have weaker states given income per capita, and while exporters of other commodities have marginally less reliable governments on average, they are not consistently weak.

additional models containing $\log(sxp)$ and different sets of control variables (F9-F11). Fearon then uses multiple imputation for both the five-year and one-year datasets and re-estimates (F12-F14). In sum, the empirical relationship between primary commodity exports and civil war onset disappears, as evidenced in p-values for sxp, sxp^2, $\log(sxp)$, and the likelihood ratio tests for the joint significances of sxp and sxp^2 across different models.

Through this process we have identified a number of core requirements for the empirical robustness feature. However, Fearon briefly touches upon alternative modeling decisions, particularly with human-guided modifications to the dataset, but chooses not to explore them further. One example involves the inconsistencies in attributing civil wars to countries; e.g., the separatist movement in Azerbaijan in 1964 (when it was part of the USSR) was coded as a civil war in Azerbaijan, while civil unrest in Chechnya in the 1990s (when it was part of the USSR and Russia) was coded as a civil war in Russia. Other inconsistencies appear in the many anti-colonial movements in the 1950s and 1960s that led to conflicts and independent nations across much of Asia and Africa, and in the 2000s in East Timor. Thus, a more thorough assessment of the finding's empirical robustness would include datasets that identify civil conflicts using different criteria (e.g. [36]).

Similarly, Fearon notes trouble in collecting economic data at the time when Collier and Hoeffler conducted their analysis, ensuring many states facing conflict were omitted from the dataset. With data now more accessible, additional conflicts can be tracked, even if the dataset is uniquely formatted. Methods can be developed to normalize the data sets in ways that ensure model compatibility.

6.2.3 Discussion. The analysis of the two articles confirmed that the tasks resulting from our top-down analysis do occur when users are developing machine learning models. The bottom-up analysis resulted in 10 additional tasks, shown in bold in Table 1: VS3, VS4, VS5, IS3, IS4, MS2, SD2, MC3, MC4, MC5. Although the two articles were selected for their diversity and their documentation of the exploration of solutions, other articles may uncover additional tasks.

We noticed that in both articles the researchers iterate over a very general pattern consisting of four stages:

A. There is first a set of tasks focused on feature selection and generation (Tasks VR1, VS2, VS3, IS2, IS3)
B. There is then a set of steps focused on selecting the model (Task MS1)
C. (Optionally) configuring the model with feature weights or parameter values (Tasks VS1, PT1)
D. There is then a request for quantities of interest and metrics (Task QC1)

The specifics of feature generation and selection vary very widely as the researchers explore different models.

Another significant observation is that there are a few tasks from our initial task analysis that did not come up in the analysis of these articles. Asking for intermediate results (QC2) did not appear in the narrative of the articles, but we are sure that the researchers did look at intermediate data. Imputation based on other variables (VS4) did not appear either, but we know this is a very common task and we expect at least the neuroscience researchers took a shortcut by removing values (IS3) instead. Augmenting the data (VS5) does not appear, but it is something that was pursued by other work in [36] mentioned earlier.

Perhaps the most interesting observation concerns how a user would build on previous solutions and models. Although the tasks that involve model and solution comparison (MC1-MC3) did not appear directly in the analysis of the articles, they are implicit in the task sequence. The neuroscience article was comparing SVM and logistic regression (MC1), using different populations (MC2), and using different features (MC3). The social sciences article leans more heavily on feature generation and selection (MC3).

Although this was not apparent in the articles analyzed, we expect that users would interact with an AutoML system by taking a previous solution and modifying it, or exploring variations of it. Therefore, we added comparison of model parameters and comparison of models (MC4-MC5). In addition, if users compare solutions or models, we would expect that they would request new solutions that are variations of previous ones. Hence, we added tasks for replacing models (MS2) and replacing data preparation steps (SD2) in prior solutions. HGML has the potential not only for bringing users the power of exploring additional machine learning models more efficiently, but allowing them to easily compare solutions in a more compact and effective manner.

7 Designing HGML Systems

Effective design of HGML systems require mapping the task analysis into requirements for the user interface and for the AutoML planner.

Table 4 summarizes user interface requirements (URe) and AutoML planner requirements (PRe) for HGML. Some tasks mapped to multiple requirements and some requirements spanning across multiple tasks.

7.1 Accommodating HGML User Interface Requirements

In our baseline system, described earlier, the focus is primarily on the construction of a solution, which entails a single interaction with the user and a single, known endpoint: a solution that includes an estimated model. In HGML, the interaction is no longer linear, as we extract domain knowledge before the model is constructed, we increase the number of models constructed and the user views those models and contributes more domain knowledge. Rather, the interaction is better characterized as iterative and nonlinear. It is iterative because models are returned, refined, and then re-estimated. It is nonlinear because a user may return to an earlier model, or go back to any associated options, after running an arbitrarily number of models with many different options selected.

Individually, the HGML requirements listed in Table 4 can be easily incorporated into our existing user interface. For example, formula builders and query constructors could be added

Table 4. **Requirements for Human-Guided Machine Learning.**

Task	User Interface Requirements	Machine Learning Planner Requirements
[VS1] The user may require that certain variables be given more priority	URe1: Allow user to specify the priority of variables to be included in the model	PRe1: Planner must use modeling primitives that accept guidance about the relative weight of different variables
[VS2] The user may suggest new variables that combine existing variables	URe2: Allow user to select features to be combined and to create a specific rule or function to generate the new values	PRe2: Planner must add a feature generation primitive based on a user-provided rule or function PRe3: Planner must include user specified primitives in the solution
[VS3] The user may modify the framing (eg temporal, spatial) in which the variables exist	URe3: Allow user to create additional data points or eliminate data points from the dataset based on time and space qualifiers	(No requirement)
[VS4] The user may fill in missing values in data based on existing variables	URe4: Allow user to define a rule or function to generate the missing values	PRe4: Planner must perform an imputation step based on a user-provided rule or function
[VS5] The user may augment the available data with new variables	URe5: Allow user to merge new variables with existing data	(No requirement)
[VR1] The user may ask that certain variables be included or excluded in a model	URe6: Allow user to select variables to be included or excluded from the model	PRe5: Planner must use modeling primitives that accept guidance about what variables need to be included in the model
[IS1] The user may highlight instances that need to be accounted for in the model	URe7: Allow user to select instances that need to be accounted for in the model	PRe6: Planner must use modeling primitives that accept guidance about instances to be included in the model
[IS2] The user may select a subpopulation based on properties of the instances	URe8: Allow user to specify parameters indicative of a subpopulation to divide the data URe9: Allow user to define a rule or function to generate a subpopulation	(No requirement)
[IS3] The user may remove specific instances (e.g., outliers, poor quality, etc)	URe10: Allow user to remove instances from dataset	(No requirement)
[IS4] The user selects training and test data, optionally with cross-validation specifications	URe11: Allow user to group dataset instances into different combinations of training and test data	PRe7: Planner must accept a specification of training and test data and cross-validation requirements
[MS1] The user may request a specific type of model	URe12: Allow user to specify the class of model desired	PRe8: Planner should use a hierarchy of primitives grouped according to model type, and include the user selected class in the solution PRe3: (same as above)
[MS2] The user may request a specific type of model to replace the model in a prior solution	URe13: Allow user to select a previous solution and indicate what new model is desired	PRe9: Planner should modify a prior solution to include the new model
[PT1] The user may request the use of specific parameter values	URe14: Allow user to specify a model and the parameter values desired	PRe10: Planner should include user specified model and parameter values in the solution
[SD1] The user may request specific data preparation steps (primitives or more general steps) to be present in the solution	URe15: Allow user to specify types of data preparation primitives to be included in a solution (i.e. data imputation)	PRe11: Planner should include a hierarchy of primitives grouped by classes according to their function, and include the user selected class in the solution PRe3: (same as above)
[QC1] The user may request quantities of interest (QI's) to be computed	URe16: Allow user to request a particular statistic test and parameters	PRe12: Planner should include primitives to generate metrics and standard statistical tests for models
[QC2] The user may specify certain intermediate results that may be of interest	URe17: Allow user to request results after any step in a solution	PRe13: Planner should return intermediate results of a pipeline
[MC1] The user may ask for two or more solutions each with a different type of model	URe18: Allow user to specify multiple models to be included in solutions that have otherwise the same steps	PRe7: (same as above) PRe14: Planner should generate solutions that use the same steps but differ in the type of model used
[MC2] The user may ask for two or more solutions with a given model but each trained on different subsets of the instances	URe11 (same as above) URe12 (same as above)	PRe6 (same as above) PRe7 (same as above)
[MC3] The user may ask to compare two or more solutions with same model but different data preparation steps	URe19: Generate comparative explanations for two given solutions	PRe15: Planner must be able to return detailed provenance records of the solutions PRe16: Primitives must support explanation
[MC4] The user may ask to compare two or more solutions with the same model but different data preparation steps	URe20: Contrast two solutions in terms of the steps involved	PRe15 (same as above)
[MC5] The user may ask for a comparison between two models, all other solution steps being the same	URe21: Generate comparative explanations for two given models	PRe16 (same as above) PRe17: Primitives must support comparative explanation

to create new variables from existing ones (URe2) and to select subpopulations (URe7, URe8). These types of interactions may be facilitated with graphical displays of the data, for example brushable density plots.

More challenging will be the *combination* of HGML requirements to reflect the iterative and nonlinear nature of the interactions discussed above. For example, the user would need to refer to specific solutions and their individual steps in relation to one another (URe13-URe17). Data may be manipulated for one model, and that manipulation may be undone for the next, only to be recalled later in the session (URe13). Users may want to compare two solutions (URe17) where one step differs (URe16). Maintaining usability and transparency, essential for user trust, becomes more complex in such interactions.

7.2 Accommodating HGML Planner Requirements

Some requirements for the planner concern the primitives available to the planner for composing solutions. For modeling primitives, these would be to accept input about feature weights (PRe1), variables to be included in the model (PRe5), instances to be accounted for in the model (PRe6), and support explanation (PRe16). The planner would need to access metadata about primitives that describes them in this light. Our planner includes a metadata catalog that is extensible to new primitive metadata. The planner would have to be extended to accept constraints that restrict the solutions it generates accordingly. Other AutoML systems, such as AutoWEKA, Auto-sklearn, and TPOT, do not have metadata in their primitives, and would have to be extended in that way. The inclusion of primitives for generation of metrics and quantities of interest (PRe12) is not difficult to accommodate, as it simply involves having the primitives.

We note that when a user defines a feature generation step (PRe2) or imputation step (PRe4) this needs to be turned into a primitive that the planner incorporates into the solution, since it would have to be applied to the test data. Our planner does not yet support this, and neither do other planners. All consider the primitive library as pre-existing.

Another set of requirements concern seeding the solutions with some partial specification, so the search is restricted to solutions which conform to that specification. These requirements include the appearance of a certain imputation (PRe4) or data preparation primitive in the solution (PRe3), the use of a certain type of model (PRe8, PRe10, PRe11), and the use of specific parameter values for models (PRe10). Our planner has a phase where it builds a solution skeleton that it elaborates in later phases. It would have to be extended to accept a skeleton formed with the user's requirements, which would not be very difficult. For AutoWEKA and Auto-sklearn, it would be relatively easy to support the use of a certain type of model, but not other requirements as they do not assemble multi-step solutions. For TPOT, which uses a genetic algorithm, it is unclear that the exploration for solutions could be controlled not to deviate from the initial skeleton.

Recording the provenance of the solutions generated would support important aspects of explanation and comparisons (PRe13 and PRe15). Our planner fully supports these.

Supporting the specification of training and test data and cross validation (PRe7) is an overarching function for solution generation. Our planner already supports this. Other planners would have to be extended accordingly.

Perhaps the most interesting extension stems from requirements about the use of a hierarchy of primitives grouped in classes (PRe8, PRe11). This would support users stating the increasingly more specific requests, for example, for "a naïve Bayes classifier", "a hierarchical Bayes classifier", and the "sklearn-hierarchicalClassification primitive". Our planner supports these hierarchies in its primitive catalog, both based on algorithm type (e.g., Bayesian vs deep learning primitives) and algorithm function (e.g., imputation vs feature generation primitives). Other AutoML planners do not currently support these hierarchies.

8 Conclusions

In this paper we presented requirements for human-guided machine learning, where domain experts use their knowledge to affect how an automated machine learning system generates models. These requirements are based on a task analysis of the different kind of interactions that would need to be supported between users and AutoML systems. We presented an analysis that reconstructs the potential interactions with such a system by the authors of two articles in neuroscience and social sciences that explore a variety of models. Our results show that users would follow repetitive patterns in their interactions. We also presented a baseline system that enables users to simply ask an AutoML system to generate a solution, and analyzed several AutoML systems in terms of how they would need to be extended to support the requirements. In future work, we plan to extend our baseline system with the requirements described in this paper. This would open exciting possibilities for domain experts to generate machine learning models of improved quality for many problems without the help of machine learning experts.

ACKNOWLEDGMENTS

This material is based on research sponsored by the Defense Advanced Research Projects Agency (DARPA) under agreement numbers FA8750-17-C-0106 and FA8750-17-2-0114. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

REFERENCES

[1] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms," in *Proc. of KDD*, pp. 847–855, 2013.

[2] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and Robust Automated Machine Learning," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., pp. 2962–2970, 2015.

[3] R. S. Olson, R. J. Urbanowicz, P. C. Andrews, N. A. Lavender, L. C. Kidd, and J. H. Moore, "Automating Biomedical Data Science Through Tree-Based Pipeline Optimization," in *Applications of Evolutionary Computation*, pp. 123–137, 2016.

[4] N. Fusi, R. Sheth, and H. Melih Elibol, "Probabilistic Matrix Factorization for Automated Machine Learning," *ArXiv E-Prints*, May 2017.

[5] AUTOML, "AutoML home page," 07-Oct-2018.

[6] Google, "Google Cloud AutoML," 07-Oct-2018.

[7] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

[8] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[9] J. Honaker and V. D'Orazio, "Statistical Modeling by Gesture: A Graphical, Browser-based Statistical Interface for Data Repositories," in *Extended Proceedings of ACM Hypertext*, 2014.

[10] Y. Gil *et al.*, "P4ML: A Phased Performance-Based Pipeline Planner for Automated Machine Learning," in *Proceedings of Machine Learning Research, ICML 2018 AutoML Workshop*, 2018.

[11] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the People: The Role of Humans in Interactive Machine Learning," *AI Mag.*, vol. 35, no. 4, p. 105, 2014.

[12] B. C. Kwon *et al.*, "Clustervision: Visual Supervision of Unsupervised Clustering," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 142–151, Jan. 2018.

[13] S. Liu, J. Xiao, J. Liu, X. Wang, J. Wu, and J. Zhu, "Visual Diagnosis of Tree Boosting Methods," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 163–173, 2018.

[14] Y. Ming, H. Qu, and E. Bertini, "RuleMatrix: Visualizing and Understanding Classifiers with Rules," *CoRR*, vol. abs/1807.06228, 2018.

[15] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer, "Wrangler: Interactive Visual Specification of Data Transformation Scripts," in *ACM Human Factors in Computing Systems (CHI)*, 2011.

[16] B. Jiang and J. Canny, "Interactive Machine Learning via a GPU-accelerated Toolkit," in *Proceedings of the 22nd International Conference on Intelligent User Interfaces - IUI '17*, Limassol, Cyprus, pp. 535–546, 2017.

[17] E. Sherkat, S. Nourashrafeddin, E. E. Milios, and R. Minghim, "Interactive Document Clustering Revisited: A Visual Analytics Approach," in *Proceedings of the 23rd International Conference on Intelligent User Interfaces - IUI '18*, Tokyo, Japan, pp. 281–292, 2018.

[18] A. Smith, V. Kumar, J. Boyd-Graber, K. Seppi, and L. Findlater, "Closing the Loop: User-Centered Design and Evaluation of a Human-in-the-Loop Topic Modeling System," in *Proceedings of the 23rd International Conference on Intelligent User Interfaces - IUI '18*, Tokyo, Japan, pp. 293–304, 2018.

[19] L. Micallef *et al.*, "Interactive Elicitation of Knowledge on Feature Relevance Improves Predictions in Small Data Sets," in *Proceedings of the 22nd International Conference on Intelligent User Interfaces - IUI '17*, Limassol, Cyprus, pp. 547–552, 2017.

[20] K. Dhamdhere, K. S. McCurley, R. Nahmias, M. Sundararajan, and Q. Yan, "Analyza: Exploring Data with Conversation," in *Proceedings of the 22nd International Conference on Intelligent User Interfaces - IUI '17*, Limassol, Cyprus, pp. 493–504, 2017.

[21] L. Sha, P. Lucey, Y. Yue, P. Carr, C. Rohlf, and I. Matthews, "Chalkboarding: A New Spatiotemporal Query Paradigm for Sports Play Retrieval," in *Proceedings of the 21st International Conference on Intelligent User Interfaces - IUI '16*, Sonoma, California, USA, pp. 336–347, 2016.

[22] N.-C. Chen, J. Suh, J. Verwey, G. Ramos, S. Drucker, and P. Simard, "AnchorViz: Facilitating Classifier Error Discovery through Interactive Semantic Data Exploration," in *Proceedings of the 23rd International Conference on Intelligent User Interfaces - IUI '18*, Tokyo, Japan, pp. 269–280, 2018.

[23] H. Dev and Z. Liu, "Identifying Frequent User Tasks from Application Logs," in *Proceedings of the 22nd International Conference on Intelligent User Interfaces - IUI '17*, Limassol, Cyprus, pp. 263–273, 2017.

[24] T. Y. Lee, C. Dugan, and B. B. Bederson, "Towards Understanding Human Mistakes of Programming by Example: An Online User Study," in *Proceedings of the 22nd International Conference on Intelligent User Interfaces - IUI '17*, Limassol, Cyprus, pp. 257–261, 2017.

[25] T. Intharah, D. Turmukhambetov, and G. J. Brostow, "Help, It Looks Confusing: GUI Task Automation Through Demonstration and Follow-up Questions," in *Proceedings of the 22nd International Conference on Intelligent User Interfaces - IUI '17*, Limassol, Cyprus, pp. 233–243, 2017.

[26] Y. Gil, J. Honaker, V. D'Orazio, S. Gupta, Y. Ma, and D. Garijo, "An Initial Task Analysis for Human-Guided Machine Learning," *ArXiv E-Prints*, 2019.

[27] H. Wallach, "Computational social science ≠ computer science + social data," *Commun. ACM*, vol. 61, no. 3, pp. 42–44, Feb. 2018.

[28] S. Athey, "Machine Learning and Causal Inference for Policy Evaluation," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15*, Sydney, NSW, Australia, pp. 5–6 2015.

[29] Y. Gil, J. Honaker, V. D'Orazio, S. Gupta, Y. Ma, and D. Garijo, "Towards Human-Guided Machine Learning: Top-Down and Bottom-Up Task Analysis," *ArXiv E-Prints*, 2019.

[30] D. Zhu *et al.*, "Classification of Major Depressive Disorder via Multi-Site Weighted LASSO Model," *CoRR*, vol. abs/1705.10312, 2017.

[31] B. Riedel, D. Zhu, N. Jahansad, and M. Harrison, "MRI based classification of Major Depressive Disorder in 16 Cohorts Worldwide: An ENIGMA machine learning study," *Rev. Mol. Psychiatry*, 2018.

[32] P. M. Thompson *et al.*, "The ENIGMA Consortium: large-scale collaborative analyses of neuroimaging and genetic data," *Brain Imaging Behav.*, Jan. 2014.

[33] T. Wolfers, J. K. Buitelaar, C. F. Beckmann, B. Franke, and A. F. Marquand, "From estimating activation locality to predicting disorder: A review of pattern recognition for neuroimaging-based psychiatric diagnostics," *Neurosci. Biobehav. Rev.*, vol. 57, pp. 328–349, 2015.

[34] J. D. Fearon, "Primary Commodity Exports and Civil War," *J. Confl. Resolut.*, vol. 49, no. 4, pp. 483–507, Aug. 2005.

[35] P. Collier and A. Hoeffler, *Greed and Grievance in Civil War*. The World Bank, 1999.

[36] N. Sambanis, "What Is Civil War? Conceptual and Empirical Complexities of an Operational Definition," *J. Confl. Resolut.*, vol. 48, no. 6, pp. 814–858, 2004.