Domain Knowledge Elicitation for Data Curation to Promote Trustworthiness in Artificial Intelligence

Mary Versa Clemens-Sewall*, Emma Rafkin†, and Christopher Cervantes‡
Asymmetric Operations Sector
The Johns Hopkins University Applied Physics Laboratory
Laurel, MD
{ *mary.versa.clemens-sewall, †emma.rafkin, ‡christopher.cervantes }@jhuapl.edu

2024 International Conference on Assured Autonomy (ICAA) | 979-8-3315-2101-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICAA64256.2024.00013

***Abstract*—Designers and developers of artificial intelligence (AI) can begin pursuing trustworthiness in an AI-enabled system long before it reaches an end user. Early in the AI lifecycle, data curation affords an opportunity for data scientists to promote trustworthiness by choosing data transformations and splits that prioritize performance in the deployed environment, not just on the data available for training. Key to enabling such purposeful data curation is the elicitation of actionable domain knowledge from various sources, including experts in the field in which the AI-enabled system will be used. In this paper, we offer a framework and set of questions for eliciting domain knowledge that drives data curation for trustworthy AI.**

***Index Terms*—trustworthy AI, data curation, subject matter expertise, knowledge elicitation, domain knowledge**

I. Introduction

In projects with strong data science components, data scientists primarily focus on developing machine learning (ML) models, generating data analyses, finding trends, and managing data [1]. The work of building models and managing data typically includes *data curation*: the process of preparing an existing dataset for use with an AI-enabled system. Data curation refers to the activities after dataset creation (e.g., collection, annotation) and before finishing model training. Data curation includes activities like data munging, data cleaning, and data splitting. As part of the data curation process, data scientists typically develop an intuition about the data and the task by synthesizing their prior experiences, their experience with the data, their reading of the available documentation, and their interactions with experts [2].

Data scientists' informally-acquired intuition about a dataset and ML task can be understood as a derivation from *domain knowledge*, or relevant information about the project's area of interest. This can include an understanding of the project goals, the environment in which capabilities will be deployed, the contexts in which analyses will be interpreted, how data was collected, and what data values mean. In practice, data scientists' derived domain knowledge is often satisficing; they may learn only what they need to perform the tasks as they understand them with the data they have available [3].

The process of collecting domain knowledge is necessary for every project, but there is not a generally accepted framework for how data scientists incorporate domain knowledge into their projects. As a discipline, data science is sometimes depicted as interacting with reliable standards and practices, but it is inherently heterogeneous: each dataset and task is different, and the domain knowledge collection process requires collaboration between data scientists and domain experts [4]. These *subject matter experts*, or SMEs, are people with extensive knowledge and skills in a certain domain [5], and though data scientists may be able to intuit some domain knowledge, collaboration with SMEs is particularly necessary when projects with data science components are not themselves about data science (e.g., business analysis, predictive health modeling).

As data scientists design and develop an AI-enabled system, SME collaboration can help foster the trustworthiness of the eventual system. Ensuring that a system meets user expectations in a deployed environment requires understanding the task, its background, the deployed environment, the data as a collection of data points, and the dataset as a whole, all of which are topics around which data scientists and SMEs can build a common understanding to facilitate project success.

The main contributions of this work are:

- A definition of the data curation phase of the AI product lifecycle, distinguishing how data is prepared for AI/ML from how data is obtained or annotated
- An actionable definition of trustworthiness for AI-enabled systems, concretely grounding data science practice amid the need for reliable systems
- A consolidated discussion of the need for and methods of practical collaboration between data scientists and SMEs, including references to example open source tools for domain knowledge elicitation and data curation
- Twelve domain knowledge questions, consolidated in Table I, to help guide collaboration between data scientists and SMEs

In Sections II and III, we define data curation for trustworthy AI-enabled systems, particularly relative to data science in practice and what is meant by data in this context. The following Sections IV, V, and VI reflect the categories of domain knowledge elicitation questions in Table I, and discuss the high level elicitation goals represented by the questions.

This work is supported by Carnegie Mellon University Software Engineering Institute.

979-8-3315-2101-1/24/$31.00 ©2024 IEEE
DOI 10.1109/ICAA64256.2024.00013

TABLE I
DOMAIN KNOWLEDGE QUESTIONS

Section	Question		
	#	*Title*	*Possible Wording*
IV	1	Task	What is the task?
	2	Current Approach	How is the task currently done? What are current practices and workflows?
	3	Current Limitations	What limitations in the current approach will be addressed by an AI-enabled system?
	4	Performance Evaluation	How will the approach be evaluated? What metrics are currently in use?
V	5	Systematic Noise	What technology was used to measure and record values in the data, and how might that technology systematically introduce noise? Put another way, are any of the features or labels imperfect proxies and, if so, how do they systematically differ from the true features and labels?
	6	Missing Values	What does a missing value indicate (e.g., not measured, not recorded, a negative result)? Is it completely random whether a value is missing, or else are some observations more likely to have a missing value?
	7	Allowable Values	What is the allowable range or set of values for each feature? Does a certain value in one feature rule out the possibility of the same observation having an otherwise permissible value in another feature?
	8	Domain-Specific Feature Engineering	Is it possible (i.e., for a trained human analyst or researcher) to determine the label (or, more generally, to perform the AI-enabled system's task) from exclusively the features present in the dataset? If so, what subsets or combinations of features would be most important to making the determination? If not, which important features are missing?
	9	Selected Characteristics	Are there dimensions of diversity in the data, either explicitly encoded in a feature or latent among the aggregate of features, along which reliable performance is key?
VI	10	Data Coverage	Is each possible input in the deployed environment similar to at least one data point in the dataset? How frequently is the AI-enabled system, once deployed, expected to encounter an input substantially different from all the available data?
	11	Covariate Shift	Are certain kinds of inputs more or less common in the deployed environment than they are represented in the data available for development? How does the true distribution of values for each feature differ from the distribution observed in the data?
	12	Label Shift	Would a feature vector, if observed in the deployed environment, have a different true label value (i.e. "right" answer for the AI-enabled system to predict) than the same feature vector is labeled in the data? For instance, in the case of a binary label, are borderline cases more likely to be labeled with a 0 in the data but a 1 in the deployed environment? Are there other such biases in the data that should not be replicated in the behavior of the AI-enabled system?

II. AIM FOR AI TRUSTWORTHINESS IN THE DATA CURATION PHASE

A common way parameters are learned from data is through ML: a model architecture is selected and curated data are used to train and validate the model's parameters. Data curation, the process of preparing an existing dataset for use with an AI-enabled system, is specifically the act of consuming one dataset and creating another from which parameters will be derived. Data curation comprises a significant phase in the AI product lifecycle [2], and we contend that data curation is not just a preparatory step for building an AI-enabled system but an opportunity to promote trustworthiness in that system.

The data curation phase is an early and requisite phase in the development of any system that incorporates an ML model. During curation, data scientists split available data into training and validation datasets. Data curation is performed after all data has been obtained and annotated and concludes with the completion of ML model training. That is, data curation may include preliminary training and evaluation of initial models because preliminary results may inform further curation decisions.

We focus on curation to support the trustworthiness of a deployed AI-enabled system. In this paradigm, a data scientist intends to build and deploy an AI-enabled system, intends that system to be trustworthy, and prepares training and validation datasets in support of those goals. Trust is complex and trustworthiness multifaceted [6], but for this paper, we scope the goal of trustworthiness to what we call an *actionable definition of trustworthiness for AI-enabled systems*:

> A trustworthy AI-enabled system must be optimized for performance on the true distribution of inputs it will encounter in a deployed environment.

This definition does not encompass all aspects of trustworthiness; we contend that it is a necessary if insufficient condition for trustworthiness. Moreover, this definition can be made actionable through data curation. That is, data curation activities can support the optimization of performance on the true distribution and thus foster AI trustworthiness.

Measures of performance vary based on the task and design of the AI-enabled system, but they should encapsulate the expectations of the ultimate users of the deployed system.

Many performance metrics are collected, described, and implemented in Scikit-Learn [7], and fairness metrics specifically are implemented in AI Fairness 360 [8]. See Section III-C for a discussion of the wide applicability of fairness metrics.

Our actionable definition of trustworthiness references the true distribution of inputs in the deployed environment. Importantly, we do not assume that the data constitute a representative sample from the true distribution. In many cases, the data are repurposed or were originally created opportunistically. This is often unavoidable. For instance, a forecasting model is necessarily trained on historical data and predicting on the future, whose distribution could differ substantially from the past.

Nevertheless, we contend that the demonstrable trustworthiness of an AI-enabled system is, in fact, constrained by the extent to which the true distribution is known. Most data curation techniques expect this knowledge to be in the form of a representative sample of data from the true distribution, which we refer to as the validation set. The validation set might be extracted from the larger pool of data available for development (the remainder of which comprises the training set), or it could have been created independently from the rest of the data by sampling directly from the deployed environment. Since the data curation phase comes after the data have been obtained, we assume that the contents of the validation set have been obtained but, possibly, have not been extracted from the data as a whole. In Section VI, we offer guidance for eliciting subject matter expertise that can be used in extracting the validation set from a larger dataset that is not representative of the true distribution yet is broad enough to encompass common inputs in the deployed environment.

There may be many reasons that knowledge of the true distribution is unavailable, including, for military applications, information security. Even so, the performance of an AI-enabled system can only be assured and optimized on known distributions. The U.S. Department of Defense has adopted the DoD AI Ethical Principles [9], including the "reliable" principle, with which our actionable definition aligns:

> "The Department's AI capabilities will have explicit, well-defined uses, and the safety, security, and effectiveness of such capabilities will be subject to testing and assurance within those defined uses across their entire life-cycles."

The true distribution of inputs in the deployed environment is part of the definition of an AI-enabled system's use. The degree to which trustworthiness (by our actionable definition) can be measured, demonstrated, and achieved is limited by the extent to which the validation set is representative of the true distribution.

III. Understand data and its limitations

In the data curation phase of the AI product lifecycle, data creation and annotation is already complete. A data scientist a) has access to a dataset, b) intends to use the dataset to learn parameters for (i.e. train) a ML model, and c) aims to promote the trustworthiness of the system by training the model on a curated version of the dataset. This section defines what is meant by *data* in this context, and how that definition relates to the work of data scientists generally and the data curation phase specifically.

A. Enumerate features and labels of data points

ML can be defined as the "creation of mechanisms that can look at examples and produce generalizations" [10]. In order to do this, the "real world" must be encoded in discrete, machine-interpretable values. When data are prepared for machine learning contexts, entities or phenomena of interest are represented as sequences of *features*, or encoded aspects of the entity or phenomenon.

A single data point is typically represented as a real-valued vector $\mathbf{d}^x \in \mathbb{R}^m$, where m is the number of features.[1] Traditionally, labeled data points consist of two parts: the features ($\mathbf{d}^x$) and label(s) ($d^y \in \mathbb{R}$).[2] For example, an image of a ship may be represented as pixel values (features) and annotated with whether the ship is a military or civilian watercraft ($d^y \in \{0, 1\}$). While it is typical for a dataset to have a single label, some datasets include multiple labels such that the values of the labels for a data point form a vector ($\mathbf{d}^y \in \mathbb{R}^l$, where l is the number of possible labels).

A label is simply an encoded aspect that an ML model is trained to predict from the features of each data point [11]. We generalize this formalization of data by combining features and labels into a single representation. In this view, data point $\mathbf{d} \in \mathcal{D}$ describes all encoded aspects of an entity or phenomenon after dataset creation and annotation. Consider data describing ships, such that each data point contains encoded information about a ship's appearance (e.g., pixel values in an image) as well as other aspects (sometimes called metadata) like its age, country of origin, and type.

In framing $\mathbf{d}$ in this way, data points can be considered without reference to a specific AI-enabled system, and a system's task determines which encoded aspects are labels. For example, one system might seek to predict ship type (military or civilian) from pixel values alone. In this case $\mathbf{d}^x$ would be the pixels, $\mathbf{d}^y$ would be the ship type, and all other encoded aspects would be ignored. Another system may seek to predict country of origin from all aspects except the pixels, and a third system might attempt to cluster similar images on the sole basis of pixels and without regard to the other encoded aspects. All such cases could leverage the same $\mathbf{d}$, but the parts of the data point that are chosen to be features or labels vary by task. This is important because many spaces where

[1] In practice, human-interpretable features (e.g., an object's length, the date of a record) can be represented as more than one value in a vector. It is also increasingly common for individual features to carry no explicit meaning at all, such as in the case of word embeddings. Therefore m should be more formally understood as the length of the feature vector, rather than the number of features.

[2] In many classification problems, real-world data is annotated $d^y \in \mathbb{N}$, where each class is represented by a natural number (e.g., fishing vessel = 0, passenger ship = 1, military ship = 2, etc.). However, we consider d^y more broadly to encompass datasets curated for other purposes, like regression-based problems.

ML is applied are data-scarce. Realistically, any data scientist must consider reusing datasets.

In summary, we formally define a data point as a real-valued vector the length of the total features and labels available after creation and annotation ($\mathbf{d} \in \mathbb{R}^{m+l}$). This generalized $\mathbf{d}$ contains all encoded aspects of the entity available during development of the AI-enabled system, which can be understood as identifying correlations between those aspects.

B. Recognize limitations of data

The choice of features that will be used in the final model and how those features should be encoded is just that: a choice. Choices are made about what to include, what to exclude, and how to reduce the complexities of reality into machine-interpretable representations [2]. A model's success hinges on these choices. Creating dataset $\mathcal{D}$—describing real-world entities and phenomena as data points—is a human-centered process in which data are not passively received from an environment but are purposefully created. In some contexts, an image may be a useful approximation of an entity. In others, it may be necessary to encode other aspects of an entity into its representation. There may also be tasks in which the data collection context itself is important to encode (e.g., the camera which took the image, the time of day). Simplifying reality into data points—encoding some aspects as features or labels—excludes other aspects.

$\mathcal{D}$ is a manifestation of choices, and $\mathbf{d}$ may not describe an entity or phenomenon relative to the goals of the AI-enabled system. Instead, $\mathbf{d}$ reflects the aspects of the real world that were chosen during the creation of $\mathcal{D}$ and the further choices of how those aspects would be encoded. That is, $\mathbf{d}$ is the entity or phenomenon from the perspective of the AI-enabled system, but that doesn't mean $\mathbf{d}$ is a good or even sufficient representation for the task. It is merely the data point that was created.

To highlight the limitations in data, we contrast a data point $\mathbf{d}$ with an ideal data point $\tilde{\mathbf{d}}$ in an ideal dataset $\tilde{\mathcal{D}}$. The ideal vector, $\tilde{\mathbf{d}} \in \mathbb{R}^{m+l+p}$, contains all possible aspects of the entity or phenomenon of interest, encoded in all possible ways (constrained to real values), such that p refers to the size of the possible feature space ($m \ll p \leq \infty$). While no dataset actually consists of idealized data points, this representation highlights the choices that are made in the construction of $\mathcal{D}$. A real entity or phenomenon is only partly represented by $\mathbf{d}^x$ and $\mathbf{d}^y$; the rest of its representation appears in the idealized $\tilde{\mathbf{d}}^p$.

This formalism is particularly helpful relative to trustworthiness and fairness in AI, as issues of relative importance, mischaracterization, or distribution shift often occur in the $\tilde{\mathbf{d}}^p$ space [12], [13]. This may be because relevant aspects are missing (and only appear in $\tilde{\mathbf{d}}^p$) or because aspects are encoded in $\mathbf{d}$ as proxy features or labels where a more direct encoding appears in $\tilde{\mathbf{d}}^p$.

When classifying images into military or civilian ships, a relevant aspect may be the presence of fishing nets ($\tilde{d}_j^p \in \{0, 1\}$). In some images, this presence may be seen in the images and thus encoded by proxy in the pixels. In others, the image may be taken from an angle which obscures nets and thus does not represent $\tilde{d}_j^p$ even by proxy. ML models trained on such images, then, may yield unexpected classification results relative to an important but only partially encoded aspect of the entities of interest.

Data curation for trustworthy AI takes cues from the true distribution of inputs in the deployed environment. The true distribution encompasses the relationships between $\mathbf{d}^x$, $\mathbf{d}^y$, and $\tilde{\mathbf{d}}^p$. To account for the choices made in dataset creation and the potential differences between the environment in which the data were created and the deployed environment, data scientists need to be aware of the distribution of features and labels (e.g., training data is 50/50 military and civilian ship images, but the deployed environment is 70/30) and how the real-world entities or phenomena represented by the deployed environment's data differ from those in the training data, which might influence input features and labels.

C. Identify selected characteristics of data points

We consider the aim of promoting AI trustworthiness, according to the definition in Section II, which partially overlaps with the aim of related academic literature at the intersection of trust, AI/ML, and fairness. This literature often references "protected" or "sensitive" attributes, typically in relation to systems in which people are represented as data points (e.g., race, gender) and aims to develop AI-enabled systems that perform without systemic bias toward or against a given group.

While the specific goal of fairness with respect to protected attributes is relevant to many mission sets (e.g., suicide risk prediction among soldiers [14]), we see it as a special case of the concept of trustworthiness as defined in this paper. If the deployed AI-enabled system is expected to perform fairly across an attribute, then part of optimizing performance entails optimizing fairness across that attribute. Rather than focus on legally protected attributes, we conceptually generalize this idea to *selected characteristics*: any aspects of the entity or phenomenon of interest which are to be considered of particular interest (e.g., the type of ship, whether the ship has fishing nets) and in need of special handling (e.g., performance must be equal for all ship types). The task and the data influence which features are selected characteristics.

Fairness is a social concept that is formulated in different ways depending on the problem. By formally defining $\tilde{\mathbf{d}}$, we can understand fairness in the context of $\mathbf{d}$ as a representation of a real-world entity. A model's performance might vary along different dimensions in $\mathcal{D}$. In order for a model to be fair, the level of performance of the model must *not* vary along features or proxies that encode a predetermined dimension of note (i.e. a selected characteristic). Given this conceptual generalization, the academic literature focusing on protected attributes offers data curation techniques that can be used for any selected characteristics. The selected characteristics (if any) would be a small subset of the features in $\mathcal{D}$. Drawing from this literature, we consider three types of algorithmic fairness [15],

- *Equal Opportunity*: The proportion of true positives should be independent of selected characteristics, given the true label.
- *Equalized Odds*: The proportion of true positives and false positives should be independent of selected characteristics, given the true label.
- *Demographic Parity*: The likelihood of a given label should not vary on selected characteristics.

Consider a dataset where each data point represents a ship such that $\mathbf{d}^x$ contains some encoded aspects of the ship and d^y reflects the country of origin. Further consider the ship type as the selected characteristic (*ShipType* $= \{$military, civilian$\}$).

A classifier satisfying Equal Opportunity, the most specific type, would yield the same rate of correct predictions for military as for civilian ships.

$$P(\hat{y}^i|\textit{ShipType}^a, y^i) = P(\hat{y}^i|\textit{ShipType}^b, y^i)$$

That is, given the true label, the probability of predicting that true label does not change in the presence of the selected characteristic.

A more general type is Equalized Odds, where a satisfying classifier would yield an equal rate of correct and incorrect predictions for military as for civilian ships.

$$P(\hat{y}|\textit{ShipType}^a, y) = P(\hat{y}|\textit{ShipType}^b, y)$$

That is, given the true label, the probability of predicting any label does not change in the presence of the selected characteristic.

A classifier satisfying the most general type of fairness considered here, Demographic Parity, would yield an even distribution over labels (e.g., United States of America, People's Republic of China) given a *ShipType*.

$$P(\hat{y}|\textit{ShipType}^a) = P(\hat{y}|\textit{ShipType}^b)$$

That is, the probability of predicting a given label should be completely independent of a selected characteristic, regardless of the true label.

These types of algorithmic fairness are framed relative to outcomes:[3] the probability of $\hat{y}$ given the selected characteristic. It is expected that this framing will be typical of real-world scenarios. Rather than a data scientist unilaterally selecting characteristics and pursuing algorithmic fairness, it is likely that a SME, decision maker, or other stakeholder has defined some selected characteristic and some notion of fairness the AI-enabled system should satisfy (e.g., equal performance on civilian and military ships), including those that are legally defined as relevant. Question 9 in Section V elicits selected characteristics.

Promoting trustworthiness—in particular, performance in the deployed environment according to performance measures that encapsulate user expectations—entails promoting fairness with respect to selected characteristics. Selected characteristics are aspects of real-world entities or phenomena that have been chosen as properties which should not impact performance in the deployed environment. A model may learn relationships between *ShipType* and the country of origin, but if a SME has determined that those relationships are inappropriate for the deployed environment—that *ShipType* should not influence predictions—then data curation to improve algorithmic fairness also supports performance.

[3]As defined here, algorithmic fairness is also framed relative to classification. While this is rhetorically convenient, fairness is not restricted to classification problems. Demographic parity, for example, may be more generally stated as being satisfied when the performance of the AI-enabled system – regardless of task or performance metric – does not vary on selected characteristics.

IV. Build common ground to understand the task in the deployed environment

According to our actionable definition (see Section II), trustworthiness of an AI-enabled system reflects the degree to which performance in the deployed environment meets user expectations. Data curation helps meet these expectations by shaping the training and validation data in the development environment with the goal of optimizing performance on the true distribution of inputs in the deployed environment. This optimization, however, requires understanding the deployed environment explicitly (e.g., what will the deployed data look like) and implicitly (e.g., what are the processes by which data is created in the deployed environment and how do they differ from the processes in the development environment). Data curation to develop trustworthy AI requires understanding both the inputs to the AI-enabled system and, since data is the product of choices (see Section III-B), the task for which the system is being designed.

A. Domain knowledge about the deployed environment

The domain knowledge pertaining to the background and context of a project can be described in part as the answers to a variation on the Heilmeier Catechism [16]–[18]:

1. Task: What is the task?

2. Current Approach: How is the task currently done? What are current practices and workflows?

3. Current Limitations: What limitations in the current approach will be addressed by an AI-enabled system?

4. Performance Evaluation: How will the approach be evaluated? What metrics are currently in use?

While some projects presuppose a task (e.g., classify relevant documents), successfully answering the first question requires a broader understanding of the task landscape, which may, in turn, shape the data scientist's understanding of the deployed environment or the chosen performance metrics (e.g., what does relevant mean, do documents need to be ranked by relevance, is there some top-k for which relevance is most important). Successfully answering the second and third questions serves a similar purpose. Even if the decision to design and implement an AI-enabled system has already been made, the data scientist can better understand the task and deployed environment by learning how it was done prior to such a system or how a new AI-enabled system will fill a gap in existing workflows. Performance is measured relative to the problem that an AI-enabled system is attempting to solve, as

seen through the eyes of users. The fourth question directly seeks measures of performance in the deployed environment, which in turn helps the data scientist ensure the system is optimized using such metrics and may lead to the identification of other performance metrics that capture other user expectations of the AI-enabled system.

Consider a system designed to classify documents into one of several categories, which was a task previously performed by human experts. Without the appropriate domain knowledge, a data scientist may design a system and evaluate according to accuracy (i.e., how many documents did the system classify correctly) and have a system that yields $X\%$ accuracy on a set of validation documents.

Domain knowledge about the task in the deployed environment, however, may reveal details like the importance of some categories over others,[4] the accuracy of human experts on the same validation data (which may be worse than $X\%$), the rate of agreement between experts classifying the same set of documents, or the contours of the task in practice (e.g., in the deployed environment, experts may first triage relevant and irrelevant documents prior to classification activities). These pieces of domain knowledge are critical in informing data curation for trustworthy AI and may also help interpret results or design models.

B. Building and maintaining common ground

The questions in Table I are useful in eliciting information about the deployed environment, but asking the questions is just one part of incorporating domain knowledge into an AI-enabled system. Data science projects are typically situated in domains outside of computer science or statistics, and the data describes entities or phenomena in the domain (e.g., chemistry, public health, military equipment). Communication gaps between data scientists and SMEs or other stakeholders are common in such contexts, even though shared domain knowledge is critical to project success [1], [19], [20].

Sharing domain knowledge between data scientists and SMEs is complicated by the quantity and diversity of collaboration. While more information may yield better results, comprehensive information sharing is inefficient [21]. It is likely not possible or desirable, for example, to transform a data scientist into a chemistry expert, or a chemist into an expert data scientist. The goal of data science and SME collaborations is not to ensure each member of a project has complete understanding of the problem space but to provide enough of a shared language to facilitate work [21]. Furthermore, it is common for conflict to arise, as SMEs may not understand "the hurdles and contributions of data scientists and vice versa" [1].

The collaboration between data scientists and SMEs can be understood to build a shared mental model, where translators or brokers help each side understand the other [1]. By contrast, we prefer to frame the collaboration as building *common ground*, or a third space, between data scientists and SMEs: a hybrid environment which can be constructed at the boundary between disciplines [21]. The common ground is a space in which each side can "compare, negotiate, and integrate goals, perspectives, and vocabularies, as well as discuss shared meanings and protocols" [21]. On common ground, data scientists can ask more nuanced or accurate questions of SMEs who can in turn provide more actionable insights to the AI development process, yielding better outcomes for the project.

The creation of common ground is a bidirectional process which allows data scientists to learn about the project domain, allows SMEs to learn about data science relative to the project, and enables both groups to understand the terms, goals, and processes of one another (e.g., via the questions outlined in Table I). This notion of common ground is intentionally distinct from the typical unidirectional framing of data science projects, in which SMEs provide domain knowledge and data scientists are consumers of it [2], [4], [19], [20].

Once established, common ground must be maintained throughout the project lifecycle [21]. System requirements may change over time, and it is possible that the initial understanding between data scientists and SMEs becomes out of sync with the needs of the project. Even if the requirements and the evaluation methods of the system remain consistent over time, however, common ground must still be maintained; as the data scientists and SMEs learn more and understand one another better, the shared vocabulary and goals must reflect these changes.

The questions given in Section IV-A invite SMEs to describe the background and context for the task in their own words. The data scientist can then clarify the words' meaning. Together, the data scientist and SME arrive at the terminology of their common ground. As the data scientist proceeds to curate the data, each step aims to improve performance on the task, as performance and the task are defined in the common ground with the SME.

V. Conduct bidirectional communication to choose data transformations

In this section, we describe questions that inform data transformations that the data scientist may take to begin curating the data, i.e., creating, from the data available, a new dataset that will be used directly for training and validating the model. While it is often the goal of deep learning to allow the models to draw connections between features, feature engineering is present to some extent in every machine learning problem [10]. Working with domain experts, data scientists must codify what is known about the data in order to facilitate data curation or model design decisions [16], [22]. Transformations include applying closed-form mathematical functions to individual features or labels, aggregating multiple features or labels into new ones, and removing data points [23]. This section describes how to ask questions about individual features and labels, including steps the data scientist can take to prepare for initial and follow-up interactions with the SME.

[4]The importance of labels relates to the interpretation of performance in the deployed environment, which is distinct from the prevalence or meaning of labels in the data, which relate to the properties of individual data samples.

5. Systematic Noise: What technology was used to measure and record values in the data, and how might that technology systematically introduce noise? Put another way, are any of the features or labels imperfect proxies and, if so, how do they systematically differ from the true features and labels?

Not all data are correct. Some commonly used technologies, including biomedical measurement devices like electrocardiogram machines, introduce noise for which there are standard denoising techniques in the literature [24]. Once the data scientist learns what technology was used in dataset creation, they can research and implement appropriate denoising techniques on the features as necessary. If the noise is predictable, it is possible that it would not even register as a statistical anomaly in some algorithms, so technology-specific algorithms might need to be employed.

6. Missing Values: What does a missing value indicate (e.g., not measured, not recorded, a negative result)? Is it completely random whether a value is missing, or else are some observations more likely to have a missing value?

In preparation for asking a SME about missingness, the data scientist can first quantify and visualize the patterns of missingness. If a systematic pattern of missingness emerges, then it is unlikely that missingness is completely random. If missingness is in fact rare and completely random, then it may be ameliorated by removing (i.e., filtering) the observations or features with missing values.

If the SME and data scientist conclude that data points with missing values must be filled in and used, the data scientist can propose and experiment with imputation techniques. Imputation is not appropriate for all cases, for example if the label is missing a value, it is unlikely that it will be beneficial for the model to impute it. Depending on the proportion of values that are missing in the dataset, the choice of imputation approach can have a large effect on the data. This is in direct opposition to the goal of imputation, which is to make the missing value *not* impact the trained model's behavior without dropping the feature or affected data points entirely. Simple imputation approaches include filling the missing values in a feature with a centrality statistic (e.g., mean or mode) and propagating nearby values for similar observations (e.g., copying last month's sales volume of the same company to this month). Where simple imputation falls short, a model can be trained to predict missing values. Regression and k-Nearest Neighbors models are more interpretable, hence easier for a SME to review, but neural networks can be used, too [23]. Scikit-learn, for example, hosts a library[5] of imputers that can easily be installed and used in any Python project [25]. After exploring the options and settling on one or more candidate imputation methods, the data scientist should present the methods to the SME, along with examples of its effect on individual observations, for the SME to review.

7. Allowable Values: What is the allowable range or set of values for each feature? Does a certain value in one feature rule out the possibility of the same observation having an otherwise permissible value in another feature?

In preparation for asking a SME about allowable values, the data scientist can compute the range or set of values for each feature across all data points in the original dataset. It may also be helpful to flag possible outliers for the SME to review. Chicco et al. and Nicholson et al. list several techniques for identifying possible outliers and even correcting them [23], [26]. Additionally, there are statistical anomaly detection and correction algorithms implemented in tools such as CleanLab [27] that can be used by data scientists in order to avoid such noise in the final dataset.

8. Domain-Specific Feature Engineering: Is it possible (i.e., for a trained human analyst or researcher) to determine the label (or, more generally, to perform the AI-enabled system's task) from exclusively the features present in the dataset? If so, what subsets or combinations of features would be most important to making the determination? If not, which important features are missing?

If the SME recommends combinations of features, the combinations could be used as inputs into the model. While there are many feature engineering approaches, the data scientist can research those used on similar data, such as data of the same type or representing the same kind of entities or phenomena.

The SME might also have insight into whether or not the features in the dataset are collinear. Collinearities can be problematic for ML models downstream of the data curation process. Specifically, regression-based models assume independence between features. Collinearities can lead to unreliable predictions in regression and are easily missed when using feature weighting metrics such as permutation importance, as these metrics only test one features' importance at a time [28]. Should a SME indicate that two features might be collinear, a data scientist can evaluate feature importance in an initial model accordingly. Tools such as the RWA Web[6] can help calculate relative importance of features [29] and reveal collinearities.

9. Selected Characteristics: Are there dimensions of diversity in the data, either explicitly encoded in a feature or latent among the aggregate of features, along which reliable performance is key?

As described in Section III-C, these could be selected characteristics or subpopulations within the data distribution across which users will expect similar performance. As an example from computational chemistry, pharmaceutical and industrial chemicals differ, but a model predicting a chemical's toxicity needs to perform similarly across both kinds.

If a selected characteristic is a feature or can be systematically derived from features (e.g., by combining multiple features in a formula), the data scientist can compute group parity metrics for each of the selected characteristics. Common metrics are described and implemented in the Aequitas tool [30].

[5]https://scikit-learn.org/stable/modules/impute.html

[6]https://rwa-web.shinyapps.io/multipleregression/

If a selected characteristic is latent among the features and cannot systematically be derived, then it may be necessary to annotate the data with the selected characteristic, which falls outside the scope of data curation and this paper.

VI. Elicit information about the true distribution

In accordance with the DoD AI Ethical Principle "reliable" [9], we argue that an AI-enabled system can only be trustworthy to the extent to which its performance was optimized for the true distribution of inputs in its deployed environment. Actionable information on the true distribution, or knowledge data scientists can elicit, is necessary for this optimization. We divide this distributional knowledge into two categories:

1) Estimated parameters of the true distribution
2) A dataset that is representative of the true distribution and may be a smaller auxiliary to the larger pool of training data

Data scientists are not assumed to be domain experts, and therefore require authoritative sources attesting that either (1) the parameters are sufficiently accurate or (2) the pre-specified set is actually representative. These authoritative sources can include SMEs, project decision makers, and well-established sources of population-level statistics (e.g., the U.S. Census [31] or CIA World Factbook [32] if the true distribution is the population of people in the United States or other countries, respectively).

One goal of eliciting distributional knowledge is to obtain a set of validation data points that is as representative as possible of the true distribution. If the data as a whole are representative of the true distribution, then splitting techniques [33] can be used immediately to select a validation set. Alternatively, a representative validation set may have been created separately from the training data (e.g., by conducting measurements in the deployed environment).

The remainder of this section is dedicated to a common scenario in which the data are not observations or measurements from the deployed environment and may not accurately represent it. Three kinds of discrepancies are elicited in the following questions: data from only part of the true distribution, data whose feature vectors offer a skewed view of the true distribution, and data whose labeling scheme differs from the truth in the deployed environment.

10. Data Coverage: Is each possible input in the deployed environment similar to at least one data point in the dataset? How frequently is the AI-enabled system, once deployed, expected to encounter an input substantially different from all the available data?

The purpose of this question is to assess how much of the true distribution *can* be represented in a validation set. For example, suppose the deployed environment is the Arctic Ocean and the available data are all from tropical seas. Some fraction of the true distribution of inputs may reflect the presence of sea ice even though none of the data points from the tropics have sea ice present. Only the ice-free parts of the true distribution can be represented in a validation set, and the performance of the system can only be measured on that fraction of the true distribution. When the representativeness of the validation set is limited, only limited trustworthiness can be demonstrated.

11. Covariate Shift: Are certain kinds of inputs more or less common in the deployed environment than they are represented in the data available for development? How does the true distribution of values for each feature differ from the distribution observed in the data?

This question aims to elicit information about covariate shift, which refers to a discrepancy between the distribution of feature vectors in the training data and deployed environment [34]. To optimize for performance on the true distribution, data scientists must compensate for any such discrepancy.

In preparation for asking the SME about the true distribution, the data scientist can fit a probability distribution to the data and compute statistics. For example, if the empirical distribution of each feature is approximately unimodal and symmetric, a normal distribution might be a decent fit and so sample mean and standard deviation are relevant statistics to compute. SciPy[7] implements methods for fitting to many probability distributions [35].

It is likely that a SME will only offer information on the true distribution of human-interpretable features. If a feature has semantic meaning, there may be records or SME intuitions on the relative frequency of each of its possible values in the deployed environment. It is less likely that such domain knowledge exists for features without clear semantic meaning, unless the same featurization scheme has been used previously on data from the deployed environment. For example, coordinates of a text or image embedding are not human-interpretable, so unless the same embedding scheme has been used widely, it is unlikely that there is information available describing the features' true distribution. If all features lack this kind of semantic meaning, eliciting information about the true feature distribution may require labeling data points with human-interpretable features for which information on the true distribution is known, a data annotation task outside the scope of data curation.

12. Label Shift: Would a feature vector, if observed in the deployed environment, have a different true label value (i.e. "right" answer for the AI-enabled system to predict) than the same feature vector is labeled in the data? For instance, in the case of a binary label, are borderline cases more likely to be labeled with a 0 in the data but a 1 in the deployed environment? Are there other such biases in the data that should not be replicated in the behavior of the AI-enabled system?

This question probes the discrepancy between the task encoded in data and the task of the AI-enabled system. In general, the AI-enabled system can only be optimized for the task that the data encodes. However, if the tasks differ in a structured way, then mitigation is possible. In the case of binary labels described in the question, the decision boundary

[7]https://docs.scipy.org/doc/scipy/reference/stats.html#module-scipy.stats

for a classifier can be chosen with the deployed environment, rather than the training data, in mind. In effect, the model can be made universally more eager or cautious to predict a label of 1 in order to calibrate it to the true distribution.

Data scientists can prepare to ask SMEs this question by identifying example data points that are close to a decision boundary. These could be found manually or by training an initial model on all data points, then feeding them back into the trained initial model as inputs and computing distance to a decision boundary. By presenting these borderline cases and their labels in the dataset to SMEs, data scientists can help elicit whether the decision boundary in the dataset matches the intended decision boundary of the AI-enabled system.

Discussing specific examples with SMEs to understand features, their labels, and their relationship relative to domain knowledge may focus on edge cases or examples prone to misclassification. This discussion, however, may be useful for understanding the data more broadly, as edge cases may reveal larger patterns when discussed with SMEs [17].

Note that question 12, unlike the other questions, is only relevant to supervised learning projects—that is, tasks for which one or more encoded aspects of the data have been designated as labels.

The Sheffield Elicitation Framework (SHELF) provides an application that is effective for eliciting true distribution parameters from SMEs [36]. The application asks each SME to estimate the true parameter and also to provide information on the uncertainty in their estimate. The interface is based on research into eliciting probabilities from experts [37].

SHELF may be useful when interacting with SMEs to answer questions in this section seeking estimates of uncertain quantities:

- The fraction of the true distribution that is substantially similar to a point in the data (Question 10)
- Each feature's true distribution, as contrasted with its empirical distribution in the data (Question 11)
- Each label's true distribution, as contrasted with its empirical distribution in the data (Question 12)

Once distributional knowledge has been obtained, the data scientist can use rejection sampling [38] to winnow the data down to a subset that matches the elicited estimated parameters of the true distribution. Thereafter, splitting techniques [33] can be used to shrink the size of the validation set to free up more training data while keeping the validation set representative of the true distribution.

VII. Conclusion

Data curation is an underutilized opportunity to promote trustworthiness early in the development of an AI-enabled system. Trustworthiness, as defined in this paper, entails optimizing an ML model for performance on the true distribution of inputs in the deployed environment. For a data scientist to pursue trustworthiness, they must understand the deployed environment, the task the AI-enabled system is intended to perform, and the inputs it is likely to receive once deployed. While a data scientist may accumulate a working knowledge of the domain through interaction with the data, collaboration with SMEs and use of other sources of domain knowledge is key to making data curation decisions that promote trustworthiness.

In addition to motivating and framing the necessary interaction with SMEs and the elicitation of domain knowledge, we enumerate twelve essential questions to elicit domain knowledge in Table I. These questions elicit why the AI-enabled system is being built (i.e., the task with which AI is to assist), how the data scientist can make nuanced decisions about data transformations, and where the AI-enabled system will be deployed (i.e., discrepancies between the available dataset and the true distribution). The questions are written to be accessible to SMEs from outside data science, and we provide specific guidance and example tools that data scientists can use to maintain common ground and facilitate bidirectional communication throughout the elicitation process.

While each dataset and each AI task is different, our twelve questions point to pieces of domain knowledge that are commonly needed in data curation. There is an opportunity to build and mature tools to better facilitate the elicitation of domain knowledge in line with each of the questions we articulate. With better collaboration between data scientists and subject matter experts, domain knowledge can be more effectively shared and acted upon.

Though more work is needed to develop new data curation approaches, link domain knowledge directly with data science workflows, and build tools to support knowledge elicitation, this paper provides a framework for understanding the intersection of domain knowledge, trustworthiness, and data curation, which is an increasingly important goal as AI-enabled systems become more integrated into commercial and government environments.

Acknowledgment

We are grateful to Tom Magelinski, Neil Otte, and Michelle Liu for reading early versions of the content in the paper and contributing to conversations about the framing of data, knowledge elicitation, and trustworthiness. We also acknowledge Amber Mills, Tim Ng, Mike Castle, Sarah Rigsbee, I-Jeng Wang, and Kay Michel for their input on the actionable definition of trustworthiness and the scope of the data curation phase.

References

[1] D. Piorkowski, S. Park, A. Y. Wang, D. Wang, M. Muller, and F. Portnoy, "How AI developers overcome communication challenges in a multidisciplinary team: A case study," *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, no. CSCW1, pp. 1–25, 2021.

[2] M. Muller, I. Lange, D. Wang, D. Piorkowski, J. Tsay, Q. V. Liao, C. Dugan, and T. Erickson, "How data science workers work with data: Discovery, capture, curation, design, creation," in *Proceedings of the 2019 CHI conference on human factors in computing systems*, pp. 1–15, 2019.

[3] A. K. Heger, L. B. Marquis, M. Vorvoreanu, H. Wallach, and J. Wortman Vaughan, "Understanding machine learning practitioners' data documentation perceptions, needs, challenges, and desiderata," *Proceedings of the ACM on Human-Computer Interaction*, vol. 6, no. CSCW2, pp. 1–29, 2022.

[4] S. Passi and S. J. Jackson, "Trust in data science: Collaboration, translation, and accountability in corporate data science projects," *Proceedings of the ACM on Human-Computer Interaction*, vol. 2, no. CSCW, pp. 1–28, 2018.

[5] A. Yarovoy, Y. Nagar, E. Minkov, and O. Arazy, "Assessing the Contribution of Subject-matter Experts to Wikipedia," *ACM Transactions on Social Computing*, vol. 3, pp. 1–36, Dec. 2020.

[6] K. Kelton, K. R. Fleischmann, and W. A. Wallace, "Trust in digital information," *Journal of the American Society for Information Science and Technology*, vol. 59, no. 3, pp. 363–374, 2008.

[7] "3.4. metrics and scoring."

[8] R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, S. Nagar, K. N. Ramamurthy, J. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, and Y. Zhang, "AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias," Oct. 2018.

[9] U. S. D. of Defense, "DoD adopts ethical principles for artificial intelligence [press release]," 2020.

[10] Y. Goldberg, *Neural Network Methods for Natural Language Processing*. Synthesis Lectures on Human Language Technologies, Springer Cham, June 2022.

[11] E. Hossain, *Machine Learning Algorithms*, pp. 117–259. Cham: Springer International Publishing, 2024.

[12] B. Frénay and M. Verleysen, "Classification in the presence of label noise: a survey," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 5, pp. 845–869, 2013.

[13] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, T. Lee, E. David, I. Stavness, W. Guo, B. A. Earnshaw, I. S. Haque, S. Beery, J. Leskovec, A. Kundaje, E. Pierson, S. Levine, C. Finn, and P. Liang, "WILDS: A Benchmark of in-the-Wild Distribution Shifts," july 2021. arXiv:2012.07421 [cs].

[14] M. K. Nock, M. B. Stein, S. G. Heeringa, R. J. Ursano, L. J. Colpe, C. S. Fullerton, I. Hwang, J. A. Naifeh, N. A. Sampson, M. Schoenbaum, A. M. Zaslavsky, R. C. Kessler, and A. S. Collaborators, "Prevalence and correlates of suicidal behavior among soldiers: results from the army study to assess risk and resilience in servicemembers (army starrs)," *JAMA psychiatry*, vol. 71, p. 514–522, May 2014.

[15] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM computing surveys (CSUR)*, vol. 54, no. 6, pp. 1–35, 2021.

[16] D. Kerrigan, J. Hullman, and E. Bertini, "A Survey of Domain Knowledge Elicitation in Applied Machine Learning," *Multimodal Technologies and Interaction*, vol. 5, p. 73, Nov. 2021.

[17] S. Park, A. Y. Wang, B. Kawas, Q. V. Liao, D. Piorkowski, and M. Danilevsky, "Facilitating knowledge sharing from domain experts to data scientists for building nlp models," in *26th International Conference on Intelligent User Interfaces*, pp. 585–596, 2021.

[18] G. H. Heilmeier, "Some reflections on innovation and invention," *The Bridge*, 1992.

[19] Y. Hou and D. Wang, "Hacking with npos: collaborative analytics and broker roles in civic data hackathons," *Proceedings of the ACM on Human-Computer Interaction*, vol. 1, no. CSCW, pp. 1–16, 2017.

[20] A. X. Zhang, M. Muller, and D. Wang, "How do data science workers collaborate? roles, workflows, and tools," *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, no. CSCW1, pp. 1–23, 2020.

[21] Y. Mao, D. Wang, M. Muller, K. R. Varshney, I. Baldini, C. Dugan, and A. Mojsilović, "How data scientists work together with domain experts in scientific collaborations: To find the right answer or to ask the right question?," *Proceedings of the ACM on Human-Computer Interaction*, vol. 3, no. GROUP, pp. 1–23, 2019.

[22] J. Kornowicz and K. Thommes, "Aggregating human domain knowledge for feature ranking," in *International Conference on Human-Computer Interaction*, pp. 98–114, Springer, 2023.

[23] D. Chicco, L. Oneto, and E. Tavazzi, "Eleven quick tips for data cleaning and feature engineering," *PLOS Computational Biology*, vol. 18, pp. 1–21, 12 2022.

[24] H. Limaye and V. Deshmukh, "Ecg noise sources and various noise removal techniques: A survey," *International Journal of Application or Innovation in Engineering & Management*, vol. 5, no. 2, pp. 86–92, 2016.

[25] T. S.-L. Community, "Imputation."

[26] B. Nicholson, J. Zhang, V. S. Sheng, and Z. Wang, "Label noise correction methods," in *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–9, Oct. 2015.

[27] C. G. Northcutt, L. Jiang, and I. L. Chuang, "Confident learning: Estimating uncertainty in dataset labels," *Journal of Artificial Intelligence Research (JAIR)*, vol. 70, pp. 1373–1411, 2021.

[28] T. S.-L. Community, "Permutation importance."

[29] S. Tonidandel and J. LeBreton, "Relative importance analysis: A useful supplement to regression analysis," *Journal of Business and Psychology*, vol. 26, pp. 1–9, 03 2011.

[30] P. Saleiro, B. Kuester, L. Hinkson, J. London, A. Stevens, A. Anisfeld, K. T. Rodolfa, and R. Ghani, "Aequitas: A bias and fairness audit toolkit," Apr. 2019. arXiv:1811.05577 [cs].

[31] "Census bureau data."

[32] CIA, "The world factbook."

[33] Z. Reitermanová, "Data splitting," 2010.

[34] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, N. D. Lawrence, A. Storkey, D. Corfield, M. Hein, L. K. Hansen, S. Ben-David, and T. Kanamori, *Dataset Shift in Machine Learning*. Neural Information Processing series, Cambridge: MIT Press, 1 ed., 2008.

[35] S. Community, "Statistical functions (scipy.stats) — scipy v1.13.0 manual."

[36] C. J. Williams, K. J. Wilson, and N. Wilson, "A Comparison of Prior Elicitation Aggregation Using the Classical Method and SHELF," *Journal of the Royal Statistical Society Series A: Statistics in Society*, vol. 184, pp. 920–940, 05 2021.

[37] A. O'Hagan, C. Buck, A. Daneshkhah, J. Eiser, P. Garthwaite, D. Jenkinson, J. Oakley, and T. Rakow, *Uncertain Judgements: Eliciting Experts' Probabilities*. Statistics in Practice, Wiley, 2006.

[38] R. D. Peng, *6.3 Rejection Sampling — Advanced Statistical Computing*.