Dataset Development

Romain Egele[*] ROMAINEGELE@GMAIL.COM
University Paris-Saclay, France, and Argonne National Laboratory, USA

Julio C. S. Jacques Junior[†] JULIO.SILVEIRA@UB.EDU
University of Barcelona and Computer Vision Center, Spain

Jan N. van Rijn J.N.VAN.RIJN@LIACS.LEIDENUNIV.NL
Leiden Institute of Advanced Computer Science (LIACS), Leiden University, the Netherlands

Isabelle Guyon GUYON@CHALEARN.ORG
University Paris-Saclay, France, ChaLearn, USA, and Google, USA

Xavier Baró XBARO@UB.EDU
University of Barcelona, Spain

Albert Clapés ACLAPES@UB.EDU
University of Barcelona and Computer Vision Center, Spain

Prasanna Balaprakash PBALAPRA@ORNL.GOV
Oak Ridge National Laboratory, USA

Sergio Escalera SESCALERA@UB.EDU
University of Barcelona and Computer Vision Center, Spain

Thomas Moeslund TBM@CREATE.AAU.DK
Aalborg University, Denmark

Jun Wan JUN.WAN@IA.AC.CN
MAIS, Institute of Automation, Chinese Academy of Sciences, China

Walter Reade INVERSION@GOOGLE.COM
Google, Kaggle, USA

Abstract

Machine learning is now used in many applications due to its ability to predict, generate, or discover patterns from large quantities of data. However, the process of collecting and transforming data for practical use is intricate. Even in today's digital era, where substantial data is generated daily, it is uncommon for it to be readily usable; most often, it necessitates meticulous manual data preparation. The haste in developing new models can frequently result in various shortcomings, potentially posing risks when deployed in real-world scenarios (e.g., social discrimination, critical failures), leading to the failure or substantial escalation of costs in AI-based projects. In this chapter, we propose a comprehensive framework for dataset development. The framework consists of several stages (i.e., requirements, design, implementation, evaluation, distribution, and maintenance), each consisting of a set of possible operators (e.g., data cleaning or data reduction). We describe the various operators in detail. Finally, we address practical considerations regarding dataset distribution and maintenance. While the framework is partially based on our experience, we aim to substantiate the steps with scientific references where possible.

Keywords: Data-centric machine learning, dataset development, data preparation

[*]. These authors contributed equally to this work.

1 Introduction

In today's digital world, large amounts of data are generated daily in various domains. Machine learning methods can utilize this data to train AI models that address or automate various tasks. As machine learning is used widely in research and industry, following the wrong procedures in collecting and processing a dataset can lead to various downstream problems when models are being trained on this data, e.g., problems with privacy or fairness. In this chapter, we present a framework that aims to help develop a dataset in a more principled way and identify core actions to be performed for better management of such a project.

As mentioned by Hutchinson et al. (2021), dataset development is not a linear process that has all detailed specifications from the start. It can be structured using an agile[1] (Chin, 2004) management methodology with core components interacting with each other as well as evolving iteratively. Figure 1 presents the framework that we propose. It is structured as a representative cycle of dataset development. One cycle is composed of five components: the requirements analysis involves the principal stakeholders and consists of defining the needs of the developed dataset; the design involves the domain expert and consists of determining how to structure the dataset and its implementation; the implementation involves data creators (e.g., data/software engineers, labellers) and consists in collecting and transforming the data to be usable; the evaluation involves data scientists and adversarial testers, consists in assessing the quality of the developed dataset concerning its requirements; the distribution and maintenance involve regulation, storage, and network experts and consist of defining the storage and accessibility of the dataset.

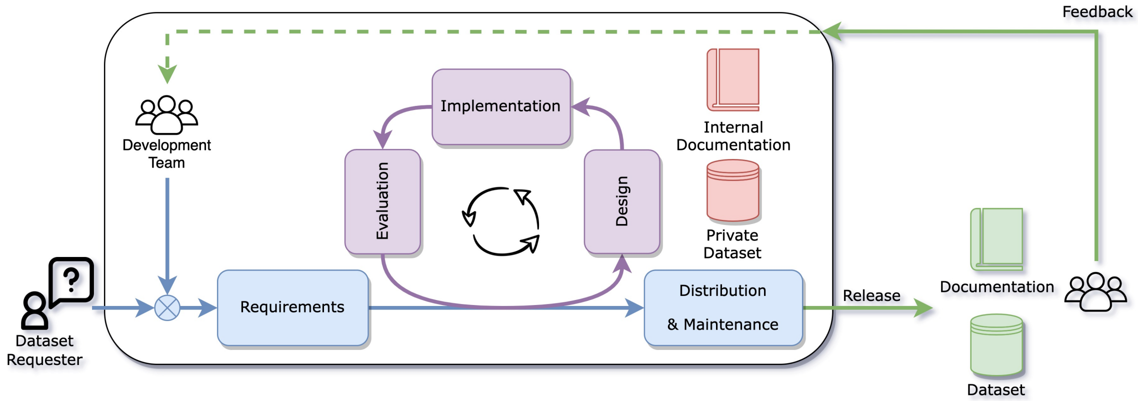

Figure 1: The dataset development cycle.

During this cycle, different aspects need to be documented. The dataset development team should keep track of internal (in the sense that it is meant to be used by the development team) documentation of common assumptions made when developing the dataset (e.g., explaining why the collected data are representative of the target population) as well as the tools or processes designed to acquire the data (e.g., survey, software). In addition, the development team should maintain public documentation explaining the content of the

[1] Agile is a common term from software development.

dataset, its purpose, and its technical usage (e.g., software to install, interface to retrieve data samples or metadata of features). This public documentation differs from internal documentation because it could contain fewer details, and sensitive information (such as personal identifiers) should be omitted. Typically, both private (raw) versions of the data and public releases of the data exist.

The framework we present aims to support the development of datasets used in machine learning. While the dataset development cycle consists of many stakeholders and roles, a small group of persons can cover multiple roles in smaller dataset development projects. We substantiated our claims with scientific literature when possible. However, some paragraphs could not be matched with such references but are the results of our personal experience (e.g., data challenges).

2 Documentation

We first review various standards of building documentation (public or private) of the development process and the final produced dataset. Documentation is essential and spans every aspect of the dataset development in order to have better transparency and traceability, which will improve trust and safety. Documentation must be performed with reflexivity, which means its goal is to clarify or uncover obscure discretionary decisions (e.g., power dynamics, differences of perception) in order to understand their effect on the produced dataset (Miceli et al., 2021). In an effort to systematize the documentation of datasets, different initiatives emerged such as new methods (Gebru et al., 2021; Bender and Friedman, 2018; Holland et al., 2018), best practices[2], and formats[3]. The main aspects addressed in dataset documentation include its purpose, composition, collection process, preprocessing, intended usage, distribution, and maintenance (Wilkinson et al., 2016).

Several standards for dataset documentation have been proposed.

Datasheets for datasets: Gebru et al. (2021) proposes specific example problems to document around the core components of the dataset development. It encourages every dataset to be accompanied by documentation of its motivation (requirements), composition (design), collection process (implementation), recommended uses, distribution, and maintenance. The NeurIPS Dataset and Benchmark track adopted these guidelines as well. Similarly, the DC-Check (DC standing for data-centric) framework was released (Seedat et al., 2022) to spawn a broader range of data-centric related tasks on a general range of applications. Bender and Friedman (2018) propose similar guidelines called “data statements”, with a focus on natural language processing applications.

FAIR principles: Wilkinson et al. (2016) focus on distribution and maintenance aspects and provide guidelines to improve the findability, accessibility, interoperability, and reuse of datasets. This initiative aims to standardize digital practices around dataset distribution and maintenance, improving the reusability of both software (e.g., data loaders, due to the use of open standards) and datasets.

[2] Metadata and data documentation: tinyurl.com/5j5ynu6p

[3] AutoML format: tinyurl.com/yc5uk4xh

Dataset nutrition label: Holland et al. (2018) follow the idea of nutrition facts labels to help create a summarized diagnosis of the quality of a dataset. It comprises diverse qualitative and quantitative modules generated through checklists or multiple statistical analyses of the dataset and displayed in a standard fashion.

However, despite the good intentions of standard documentation practices, they often must be refined for the specificities of the target use, and some attributes may not be allowed to be disclosed or stored due to data regulation (e.g., race, belief), which therefore prohibits public verification of some statistical properties.

Use Case: FACIAL EMOTION RECOGNITION

We give an example of potential aspects to be documented when developing a video dataset for facial emotion recognition (Corneanu et al., 2016). In this context, videotapes are recorded from a group of participants.

- *What is the dataset intended to be used for (e.g., for which application and for which population)? – **Requirements***
- *How are participants recruited for video recording (e.g., gender, age, hairstyle, use of glasses)? – **Design***
- *How is the privacy of recorded participants managed? – **Design***
- *For the recording protocol: Are participants following a script during recordings? Are participants stimulated by a particular incentive during recording? – **Design***
- *For the annotation protocol: How are the annotations defined (e.g., categorical, discrete, or real values)? How are annotators selected (e.g., gender, age, etc.)? How are data to be annotated defined (e.g., per frame, per video segment, with or without sound)? How many annotators observe each data? – **Design***
- *How are acquired data transformed (e.g., calibration)? How are final labels computed? What is the definition of frames and sample rate? – **Implementation***
- *For the investigation of social bias one could consider documenting some sensitive information regarding participants and annotators (e.g., gender, age, race) while being cautious to respect data regulation, privacy, and consent. What are the possible biases from the selected populations (participants, annotators)? How is the annotators' agreement evaluated (e.g., metric), and what is its value? – **Evaluation***
- *Where are the data hosted? How can the data be accessed? – **Distribution & Maintenance***

Finally, we give some practical advantages of good dataset documentation. The dataset development team can benefit from the following aspects: a better management of the dataset development by an improved understanding of why, how, and what is done to produce the dataset; a better traceability of possible bugs or flaws; an improved reusability of developed tools due to clear documentation and principled development; a reduced presence of flaws in the produced dataset; a better dataset quality. On the other hand, the dataset users can benefit from the following aspects: an improved usability of the dataset; an improved understanding and trust of the dataset; a better quality of machine learning models; and a better reporting of possible flaws discovered in the data.

Nevertheless, while documentation helps to improve dataset quality and, therefore, the quality of machine learning models using them, some limitations still exist. Companies often regard some of the information that could or should be documented as confidential, especially if it involves details about the intended product or if some of the processes involved in producing the dataset are a strategic advantage (Miceli et al., 2021). For this reason, we differentiate private (required for audits) and public documentation (required for the user). Documentation is often seen as time-consuming work that is likely to delay the completion of other tasks that are perceived as more important. It is often perceived as an optional, nice-to-have but not must-have component, and therefore, in such cases implemented last (Miceli et al., 2021). Lastly, producing complete but synthetic and clear documentation is challenging. The documentation format may vary for the different stakeholders (engineers, statisticians, business analysts) and create redundancy (pdf document, book, website). We encourage dataset development teams to use tools such as Sphinx[4], ReadTheDocs[5] and Pandoc[6] to automate the build of documentation and navigate between formats.

3 Requirements

The requirements analysis is the first step of the dataset development cycle (see Figure 1), where the dataset requester (representing the main stakeholder requiring the dataset) and the dataset development team (representing who is in charge of producing the dataset) meet to define the requirements. During this phase, the following topics can be addressed:

1. Why is the dataset needed?

    - **Application scenario:** What are the intended purposes and use cases?
    - **Machine learning tasks:** What type of machine learning techniques (e.g., supervised, unsupervised, reinforcement learning) is planned to be used? How will tasks be carved out of data?
    - **Users:** Which group of users do we expect to use the dataset?

2. How is the dataset developed?

    - **Prior work:** Are there already existing datasets filling this need (see Section 4.3 and 5.1.1 about potential sources of already collected data)? Will collecting (more) data solve the problem at hand or help in understanding it better?
    - **Method:** What dataset development protocol is planned to be used?
    - **Ethics:** Are the intended purposes ethical? What are possible fairness and privacy issues, and how will they be evaluated? Is collecting such data considered experimenting on human subjects?
    - **Risks:** What adverse usage could be done from the dataset? Are the risks worth the trouble? How can the risks be reduced?

[4] www.sphinx-doc.org
[5] readthedocs.org
[6] pandoc.org

- **Constraints:** What are the anticipated difficulties limiting the development of the dataset (e.g., recruitment of subjects)? In the case of data involving human subjects or a legal entity, it is crucial to follow regulations from governmental regulations such as GDPR (Voigt and Bussche, 2017). Sometimes, the data needs to be anonymized, cannot be stored anywhere (risk of data leaks), and its legal framework (e.g., application, lifespan) needs to be defined before acquisition.
- **Development team:** Who is composing the dataset development team? Who will lead the effort? How are the roles distributed (design, implementation, evaluation, distribution, and maintenance)?
- **Resources:** How many resources will be required to complete the whole dataset development cycle, including compensation of staff, recruitment of volunteers, payment of annotation services, computational resources, etc.? Also, the environmental impact (CO2 emissions) should be considered.

3. What is the dataset expected to be?

    - **Content:** What information does the dataset needs to contain (e.g., features, annotations, metadata)?
    - **Baseline:** What baseline modeling methods will be used to evaluate the quality of the dataset? What evaluation measures will be used, including utility, fairness, and privacy?
    - **Ownership:** Who will own the rights? Who will be legally responsible?
    - **Storage:** Where does the dataset need to be hosted?
    - **Distribution:** How is the dataset going to be accessed (e.g., through a web API)?

4. When is the dataset expected to be delivered?

When designing a dataset, one should always keep its envisioned purpose in mind. Also, the interplay between the dataset and the machine learning method is essential. A dataset represents a snapshot of the real world used to train (and possibly evaluate) a learning algorithm on a particular task. It is essential for quality assurance to involve baseline modeling methods and their performance evaluation early in the dataset development process. If the utility, fairness, ethical concerns and privacy of the data cannot be assessed in the context of an actual learning task, the dataset will likely be useless.

Regarding ethical considerations, having a committee that includes diverse members in terms of competence, demographics, and cultural backgrounds is advisable. The committee should include persons competent in the target application area, machine learning/data science, and persons representative of the subject and target population (if human subjects are involved in data collection or are affected by data usage). Additionally, it is advisable to include an ethics expert and a law expert. It is advisable to ensure that at least one member is not affiliated with the organization creating the dataset and that no member has

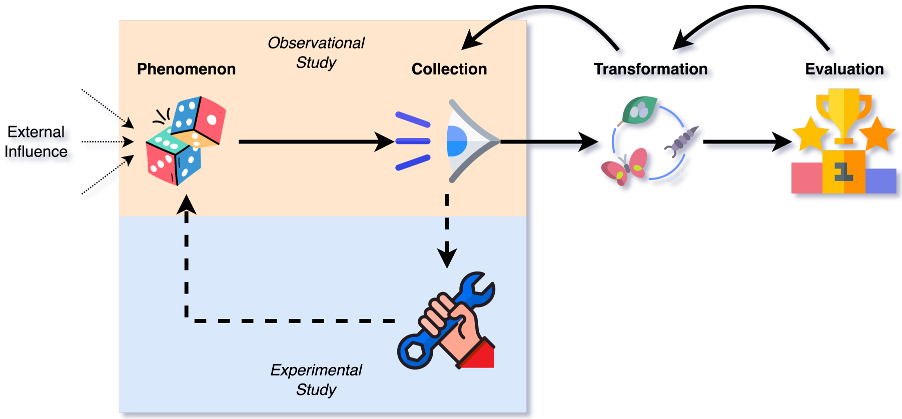

Figure 2: Illustration of observational vs. experimental studies.

a conflict of interest. In the US, such committees are called Institutional Review Board[7] and can be registered officially. Also, some general guidelines[8,9] can be followed.

4 Design

The design phase is about defining more precisely what the dataset should contain and how it will be implemented (i.e., collected and transformed), evaluated, distributed, and maintained. The complexity of implementation design, which comprises data collection and transformation, is directly related to the decision between creating a dataset from scratch or reusing, repurposing, and recycling existing data. For many use cases, processes describing dataset development already exist and are published, such as at the NeurIPS Dataset and Benchmark track[10]. We recommend exploring this literature to look for dataset development-specific methods.

The design of a dataset is often time-intensive while being crucial to innovation. Of course, designing a dataset results directly from the definition of requirements, but it is also possible to update the requirements based on insights from the design step. In fact, any critical aspect missed at the requirements and design stages may compromise the whole dataset development process. Even though these steps should be designed thoroughly, not everything can be foreseen, and flaws will likely be discovered later. For instance, consider the case where data are planned to be collected through an online survey; the development team could forget to ask participants to sign an agreement. Such an agreement can include the rights to process and use the data or to transfer the rights (including copyright). The collected data cannot be used without such an explicit agreement with the participants. Therefore, we encourage the development team to bootstrap the whole development cycle on a small scale to help refine requirements and design. As a result, the dataset development should not be a linear closed loop but an iterative and interactive process.

[7] Institutional Review Board: https://tinyurl.com/mvpd292w
[8] Ethics Guidelines for Trustworthy AI: https://tinyurl.com/2zc7hfdz
[9] Recommendation on the Ethics of Artificial Intelligence: https://tinyurl.com/3ew8xp4e
[10] NeurIPS Datasets and Benchmarks track: https://tinyurl.com/37u4cbx3

When developing a dataset, either observational (i.e., the variables of the phenomenon cannot be controlled) or experimental (i.e., the variables of the phenomenon can be controlled) data can be used (Figure 2). Similarly, data can be collected *de novo* (i.e., from scratch) or from existing sources (i.e., reuse, repurpose, and recycle).

4.1 Data Leakage

Before diving into possible designs, we emphasize the critical importance of vigilance against data leakage, which can undermine the integrity of the evaluation process. Unlike typical software bugs, data leakage can irreparably compromise months of model development. Given the myriad ways data leakage can occur, it is prudent to assume that the initial dataset may contain some form of leakage introduced during its development (Figure 1).

Participants in benchmarks and competitions often have diverse motivations beyond scientific advancement, such as monetary incentives and prestige. Some may adopt a "hacker" mindset, prioritizing circumvention over problem-solving, which necessitates the establishment of clear competition guidelines to discourage cheating.

Data leakage generally refers to the inclusion of illegitimate information in the dataset used for model development and selection (Kaufman et al., 2012). Illegitimate information pertains to data that will not be available once the model is deployed in its final environment. This leads to biased model selection and frequently results in overestimating generalization performance. It also affects the ranking of compared models in unexpected ways; for instance, the selected best model may exhibit outstanding performance in the training environment, but it could perform mediocrely in the final environment.

In this chapter, we will not address "adversarial" aspects of data leakage, such as deliberately using the ground truth targets of a test set for training, as it relates more to software security. Simply put, we focus on issues that are difficult to detect, even for someone willing to avoid leakage and who wants to follow the spirit of the task associated with the dataset.

Use Case: Leakage while collecting pictures of animals

Assume a team is organizing a challenge to predict whether an image depicts a cat or a dog. They deliver the data in two folders: one containing all cat images and the other containing all dog images. What types of data leakage should be considered?

Timestamps of the images could leak information if the data were collected on different days. For example, if the person collecting the data spent one week photographing cats and the following week photographing dogs, or if they downloaded all the cat images first, followed by the dog images. It is a good practice to randomize (in a repeatable and deterministic manner) how the files are stored.

If new photographs were taken specifically for the competition, different individuals might have used different cameras preferentially for cats or dogs. Consequently, metadata indicating the camera type (either through actual metadata embedded in the image or a proxy such as resolution and color balancing) could predict the target, thus constituting leakage. Tools like EXIF viewer[11] *can be used to inspect image metadata.*

It is generally advisable to strip files of any associated metadata, especially with images, but more is needed. A camera model can often be inferred from a raw image without using embedded metadata through features such as resolution or the specific way the JPEG is created from the camera sensor data.

[11] EXIF Viewer: https://exif.tools/

Therefore, images should be presented in a standardized format that minimizes the possibility of such leakage, or cameras should be randomized to take the pictures.

The situation becomes even more complex in scenarios such as medical imaging competitions, where it may be infeasible to eliminate the effects of varying imaging equipment (e.g., scanners). In these cases, conducting a thorough risk/benefit analysis is crucial to determine whether to include data from different imaging equipment and what metadata to incorporate. In addition, proper model analysis will need to be conducted to ensure that it did not simply learn an artefact of the sensor.

4.2 De Novo Data

This section describes design aspects for *de novo* data. The first setting to consider is which variables of the target phenomenon can be controlled or not and in which quantity they appear (e.g., difference between pixels of images and tabular dataset). We distinguish between an experimental case (where variables can be controlled) and an observational case (where variables can not be controlled). In the experimental case, when only a few variables are available, one can decide to discretize them and explore all possible combinations. When more variables are available, one often resorts to random sampling. In the observational case, even though variables cannot be controlled and impacting factors are often intractable (e.g., describing a patient in healthcare), random sampling is also advocated (random trials) to vary these factors. In this case, it is essential to check that the observed population is randomly sampled and not biased (e.g., selection bias). Other considerations for the dataset design are:

Data quantity: How many samples should the dataset have? An example of character recognition is proposed in (Guyon et al., 1998). However, it may be challenging to have such information beforehand. It is possible to refine the quantity needed by involving the modeling process during development (i.e., baseline in the evaluation step). A default choice is to collect as much data as possible and leave the choice of quantity to the user. An other possibility is to use the Hoeffding bound[12] (Burges, 1998; Bousquet et al., 2003) to estimate the number of samples needed to reach a certain level of confidence in estimating the generalization of a model. Finally, learning curves of machine learning algorithms (Mohr and van Rijn, 2022, 2023) can also be used to extrapolate the quantity of samples needed in order to reach a certain accuracy level.

Data balancing: How to make sure that you have enough samples of each group that should be represented? For instance, depending on the application, one may want to balance groups by gender, age, or educational background. It is advised to pay attention to the possible need to consider cross-sectional groups' gender × age × educational background. Unfortunately, the larger the number of grouping factors to be considered, the larger the number of samples to be collected to adhere to a minimal group size.

Data annotations: Are labels required? Do we need more advanced annotations, such as bounding boxes around the subject? If yes, how is the labeling process operated? For

[12] Learning Theory, J. Domke: https://tinyurl.com/yc2k99w4

example, this can be achieved through crowd-sourcing, citizen science, or commercial parties. When making use of such services, one should always carefully check the conditions under which these labeling operations are being performed (e.g., are the persons performing the labeling doing this under fair work conditions) and whether this process can be (semi-)automated.

Data representation: How are data represented? Many data structures exist to represent data, it can be tables, images, videos, text files, and graphs. The data can be compressed or not. The data can be accessed incrementally or all at once (we refer the reader to the HDF5 open format and library[13]). Can we provide a data reader? If tabular data are collected, what are the features to collect?

Metadata: What metadata can be collected (e.g., date of recording, operator name, temperature)? Generally, the more metadata, the better. Metadata can help identify bias or spurious correlations (potentially resulting in data leakage). The metadata explains the context of generated data, and therefore it can help determine if the data was well collected in diverse settings (e.g., at different times or temperatures). Also, metadata should not be correlated with the predicted variable. If this is the case, then some spurious correlation can be identified and resolved by modification of the dataset creation process.

4.3 Reusing, Repurposing, and Recycling Data

As discussed before, *de novo* dataset development can be time-intensive. However, many datasets are now publicly accessible (Koch et al., 2021) with a license allowing to use them free of charge (e.g., Creative Common[14]). In addition, search engines (a short list is available at the end of this section) can help find datasets corresponding to specific needs. Therefore, before performing *de novo* data collection, it is essential to investigate such opportunities, which can help save considerable resources. We refer to such methods as data reusing (analogously to reusing a plastic bottle of water by refilling it), data repurposing (analogously to repurposing a plastic bottle of water to collect leaking water from a pipe) and data recycling (analogously to breaking down the plastic bottle of water to make plastic boxes). These concepts are ordered from fewer to more modifications applied to the pre-existing data. However, the boundaries among them are not clear and can overlap.

On the one hand, data reusing is the practice of directly reusing data (i.e., without modification, including its purpose) when the use case is similar, and the required information is already available. Hence, we talk about data reusing when the initial product (i.e., the data) does not need to be transformed before being ingested and the intended purpose remains the same.

In other cases, one may reuse the same data set but repurpose it from being used in research to commercial applications. In this case, many ethical and privacy concerns must be revisited, among other aspects.

When leveraging existing data, it is important to be careful with possible deprecation, bias, and retirement of the source data. A dataset becomes deprecated when it is still

[13] https://www.hdfgroup.org/solutions/hdf5

[14] https://creativecommons.org

publicly accessible, but for some reason (e.g., social bias), it should not be used in practice. An example of a deprecated dataset is the Boston house prices dataset (Harrison Jr and Rubinfeld, 1978) is kept public for scientific traceability, reproducibility, and social bias study but discouraged from being used otherwise. In this case, the deprecation was due to social bias in the data. Similarly, some datasets can be retired (i.e., removed from public access) such as the Tiny Images dataset[15] (Torralba et al., 2008) due to the presence of derogatory terms as categories and offensive images.

In some cases, light data transformations are required to enable data repurposing. For example, a re-annotation process may be performed if the to-be-predicted target variable changes. In other cases, pre-existing data can be completed by new samples and features. For instance, the "First Impressions V2" (Escalante et al., 2017) dataset is a typical example of repurposing. The original "First Impressions" (Ponce-López et al., 2016) dataset was annotated with Big-Five personality traits with the purpose of analyzing personality from audio-visual data. In version V2, the audio-visual data was kept the same. However, new labels, such as an additional interview variable and transcriptions, were included to enable research on explainable machine learning.

Finally, one can choose to recycle a data set. Data recycling leverages existing material with the possibility of reshaping it entirely. In this case, the dataset's purpose can further differ from its initial purpose. Therefore, it has to be verified that it is allowed by the dataset's license. While data recycling typically requires less effort than *de novo* collection, these additional checks still incur an additional workload not to be underestimated. An example of recycling is the creation of an image dataset with a single face per image per sample given a pre-existing image dataset containing single or multiple people on each image with full or partial bodies (Agustsson et al., 2017).

Examples of Datasets Search Engines and Providers are Dataset Search from Google, Kaggle, OpenML.org (Vanschoren et al., 2014; Bischl et al., 2021), UCI Machine Learning Repository, Hugging Face Datasets, and the NeurIPS: Datasets and Benchmarks track.

5 Implementation

This section focuses on the implementation of the dataset development. We consider this stage to be a set of processes; Figure 3 overviews these. This figure categorizes all processes into two categories: Collection processes (blue) take as input a design (Section 4) and as output a dataset (of an arbitrary size). Transformation processes (yellow) require a dataset as input and will output a transformed version of this dataset. Some processes (green) fall into both categories.

These processes are the functions that will produce a dataset on which machine learning models are trained. Data **collection** (Section 5.1) entails the *gathering*, *acquisition*, *synthesis/generation*, and *annotation* of data. To avoid any confusion with other works that interchangeably use these words, in our chapter we called *collection* the larger class that includes all the others.

Data **transformation** (Section 5.2) includes *cleaning*, *reduction*, *representation*, and *normalization/calibration* of data. The tasks of data *integration/fusion*, and *augmentation*

[15] Tiny Images dataset: https://tinyurl.com/2vfa4xve

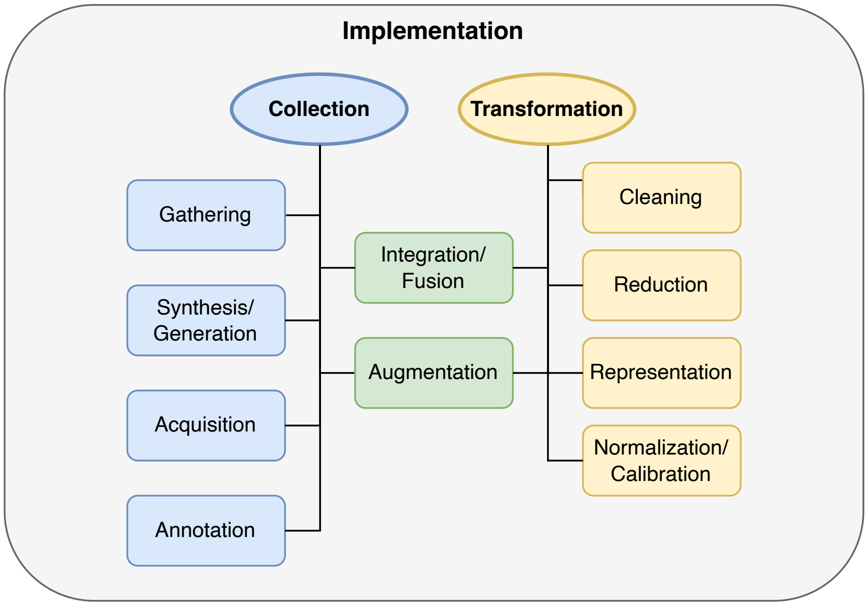

Figure 3: Categorization of sub-tasks included in data collection and transformation. Collection operators (blue) take as input a design and as output a dataset (of an arbitrary size). Transformation operators (yellow) require a dataset as input and will output a transformed version of this dataset. Some operators (green) fall into both categories. A typical data development process combines several operators into a pipeline, always starting with a collection operator.

can be categorized as both collection and transformation operators; therefore, we link them to both in Figure 3.

The implementation phase typically contains various sub-processes defined by the design. An example of such a pipeline of processes is visualized in Figure 4, in which a pipeline combines five sub-processes.

Note that data evaluation can also be considered in the case of a dynamic data collection process, for example, to keep collecting/augmenting data until a pre-defined objective is satisfied. Section 6 details on these evaluation processes. Next, we discuss each part (collection and transformation) in more detail.

5.1 Data Collection

Data collection is the set of processes that can gather, generate, or measure information in order to create a dataset. Therefore, data collection includes data gathering, data synthesis, data acquisition, and data annotation.

Any of these processes can be performed in an observational or experimental setting. In the observational setting, the investigator responsible for data collection does not interfere with the phenomenon. The distribution of collected samples is supposed to reflect the natural distribution of data. For example, a naturalist studying wildlife may set up a camera

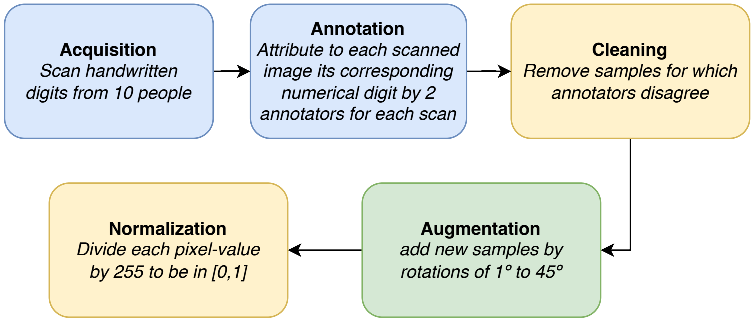
Acquisition
Scan handwritten digits from 10 people
Annotation
Attribute to each scanned image its corresponding numerical digit by 2 annotators for each scan
Cleaning
Remove samples for which annotators disagree
Augmentation
add new samples by rotations of 1° to 45°
Normalization
Divide each pixel-value by 255 to be in [0,1]

Figure 4: An example flow chart diagram of a data implementation pipeline that creates a dataset for handwritten digits classification.

trap in a forest to take pictures of animals living there. In contrast, in the experimental setting, the investigator may interfere with the phenomenon to achieve desired effects. The distribution of samples collected follows an experimental plan or design. For instance, a pet food company may want to study the influence of certain dog foods on certain dog breeds and conduct a trial, assigning different regimens to dogs of various breeds and then evaluating their energy by videotaping them. There also exists in-between cases in which data are observational, but the investigator samples data and features in an active way (Settles, 2009). For example, a photographer who gathers data by going on a photo safari may use aesthetic criteria to make their shots.

When collecting data, metadata is often available or can be created to describe processed data. It is essential to save as much metadata as possible to perform better data evaluation later.

5.1.1 Gathering

Data gathering is the set of processes by which data are brought together from sources where the data is already stored digitally, and the acquisition (Section 5.1.3) process cannot be influenced (see, e.g., Ullah et al. (2022)). The emergence of the modern web, where one can quickly access a massive amount of public data, has made the cost relatively low. These techniques suffer from high noise (e.g., picture wrongly tagged and therefore wrongly returned by a search engine) as well as ethical and regulatory limitations (e.g., privacy, copyrights, license).

An obvious way of gathering data is by using a search engine (e.g., Google Image, Bing), in which case it is critical to comply with the regulations. Another way is through web scraping, which is the practice of automating the search and downloading of data from the Internet. It provides more granularity to develop the gathering process. Tools such as Scrapy[16] can help to set up such a process. However, it must be performed responsibly, such as by following robot policies[17], which in some cases restrict access to robots.

[16] https://scrapy.org

[17] How to write and submit a robots.txt file: http://tinyurl.com/a9tvx6n8

Last, the gathering can be performed through crowd-sourcing (Roh et al., 2021; Garcia-Molina et al., 2016) where human workers are generally given micro tasks to gather bits of data that collectively become the generated dataset. More generally, it can be defined as the process of obtaining needed services, ideas, or content by soliciting contributions from a large group of people (Zhang et al., 2016). While crowd-sourcing can be applied both to data gathering and annotation, the methodologies applied differ, and its application to data annotation is detailed in Section 5.1.4. Crowd-sourcing to gather data can be performed implicitly or explicitly (Garcia-Molina et al., 2016). It is implicit when people are unaware of it, such as through website analytics. For example, by watching a movie, an individual provides data about the popularity of this movie. Then, the gathering is explicit when subjects get a request for information. The machine learning community has been using crowd-sourcing as a means to gather massive amounts of data, which can be considered a prominent way to outsource work to people reachable online. One of the most popular platforms for crowd-sourcing in machine learning at the time of writing is Amazon Mechanical Turk[18], where tasks are assigned to remote workers, which are compensated when the task is completed (Roh et al., 2021). Of course, the dataset collector also has the responsibility of checking that the workers who perform this labeling are doing so under ethical work conditions.

5.1.2 Synthesis and Generation

Data synthesis, also known as synthetic data generation, is the process of generating artificial data that mimics real-world observations and can be used to (pre)train machine learning models when actual data is difficult or expensive to get. It has become an attractive field of study because of data-hungry technologies such as deep learning (Nikolenko, 2021). Synthetic data can be generated procedurally (Queiroz et al., 2010; Wood et al., 2021) (i.e., predefined set of rules), through simulations (Dosovitskiy et al., 2017) or using generative models (e.g., generative adversarial networks (Karras et al., 2019)). One of the main motivations behind its use is the reduced cost of creating larger datasets where factors of variability (i.e., parameters of the synthetic generator) can be manipulated on demand. In addition, the generated data can often be directly annotated by leveraging the parameters of the data generator or using post-processing techniques. Other advantages include the possibility of better-controlling privacy (Yale et al., 2020; Kuppa et al., 2021) and fairness (Bhanot et al., 2021, 2022). However, one of the main limitations of synthetic data is that simulated phenomena often need to represent real-world scenarios closely, which is hard to achieve. Another challenge is related to transfer learning (Weiss et al., 2016) and domain adaptation (Ajith and Gopakumar, 2023), since a predictor trained on synthetic data (source domain) must generalize well to real-world data (target domain). Hence, the data used to eventually train the predictor implicitly should have similarities with the target data on which the trained model is deployed. Artificial data (available in large quantities) are frequently used to pre-train the model, which is later fine-tuned to real-world data (available in smaller quantities) before deployment.

The level of abstraction and realism can vary depending on the application domain, context, and needs. For example, in the case of synthetic data generated through simulations

[18] https://www.mturk.com

and computer graphics applied to autonomous driving cars (Dosovitskiy et al., 2017), the level of abstraction and realism can be associated with the rendering quality, but also with respect to the behavior of the different simulated agents and phenomena (e.g., interaction between agents, traffic, weather, etc.). Each has a particular impact on the outcomes if the data are used for training a machine learning method. The associated costs and resources (e.g., data experts, designers, software engineers, etc.) required to achieve the desired level of abstraction and the trustworthiness of the generated data are additional barriers that might limit broader usability to train and evaluate machine learning models.

Synthetic data is also reported in the literature to perform data augmentation, combining synthetic and real-world data. However, a recent study on face analysis proposed to train their models with synthetic data only (Wood et al., 2021), opening up new approaches to better address fairness and privacy.

The synthetic data generation process can introduce artifacts, including some that may leak ground truth information. An example of this occurred during the SETI Breakthrough Listen[19] challenge. In this competition, the processing date of the files (the timestamps) leaked information about the ground truth target.

5.1.3 Acquisition

Data acquisition is the process that converts a real-world signal into a digital representation (Emilio, 2013). A basic example is the acquisition of temperature data through a thermometer. Some techniques such as quantization (Gray and Neuhoff, 1998), signal sampling (Higgins, 1996) or real number encoding [20], which have already been studied in depth in the information theory literature for physical signals, are often used for this purpose.

Performing data acquisition usually requires a well-designed experimental protocol (Section 4). In addition, it is always important to record additional metadata. Some basic metadata are the measurement time, location, and model of the device used for acquisition. Metadata can be descriptive, such as an overall description of the dataset and variables (e.g., units of physical quantities, and preferably following international standards). They can also contain copyrights or license terms. In general, metadata is to be understood as data used to describe the dataset to maintain tractability about how the data were acquired. In many cases, they can become the dataset of another dataset, for example, in the case of integration/fusion. In fact, metadata play an important role in identifying bias.

5.1.4 Annotation

Data annotation is the process of mapping existing data samples to other data. It is often performed in the context of supervised learning, which includes two principal variants: classification and regression. In classification tasks, the goal is to train a model that returns one of many possible classes for each sample. In this context, data annotation is referred to as data labeling. In regression tasks, the goal is to train a model that predicts a real number given a sample. Therefore, annotating with continuous variables is much more complex than a set of fixed classes (Roh et al., 2021), which can explain why data annotation research has primarily been focused on data labeling for classification. Other types of supervised tasks

[19] SETI Breakthrough Listen Kaggle challenge: https://tinyurl.com/mrv9vtrk

[20] https://ieeexplore.ieee.org/document/8766229

exist, such as: (i) describing the title of images (image captioning), (ii) labeling the style of a music record (classification), (iii) predicting the age of an individual (regression), (iv) rating an Amazon product by a score (categorical regression), or (v) drawing a bounding-box on an object in an image (object detection).

Roh et al. (2021) propose the following categories for understanding the data labeling landscape: crowd-based labeling (e.g., via crowd-sourcing or active learning (Tang et al., 2021)), automatically labeling from existing labels (e.g., through semi-supervised learning (van Engelen and Hoos, 2020)), and the use of weak labels (Ratner et al., 2017) (i.e., generating imperfect labels, but in large quantities to compensate for the lower quality).

First, crowd-sourcing techniques (Zhang et al., 2016) are generally focused on running tasks with many workers who are not necessarily experts (either on labeling or on any particular task). Therefore, different solutions have been proposed to collect more accurate and trusted labels (e.g., see (Nowak and Rüger, 2010; Maroto and Ortega, 2018)). In this line, a general procedure for controlling and ensuring the quality of data labeling is to have multiple workers annotate the same sample so that an agreement level can be computed. This way, any bias the workers may have can be identified and mitigated, with the cost of increasing time and resources. However, it does not necessarily include human perception bias, which is much more challenging to identify and mitigate (discussed in Section 6.4). Although various inter-annotator agreement measures exist (Checco et al., 2017) for simple categorical and ordinal labeling tasks, relatively little work has considered more complex labeling tasks, such as structured, multi-object, and free-text annotations (Braylan et al., 2022). Providing adequate instructions and the proper labeling interface is also a critical success factor (Roh et al., 2021).

Then, active learning focuses on iteratively selecting the most informative (according to some pre-defined measure) unlabeled examples for the model to reduce the need for human labor, which can then be outsourced or crowd-sourced (Roh et al., 2021). The workers are expected to be accurate; thus, the key challenge is to choose the proper examples given a limited budget. In addition, semi-supervised learning can complement active learning (Gu et al., 2014; Camargo et al., 2020) by finding the predictions with the highest confidence and adding them to the labeled examples (as pseudo-labels). In contrast, active learning can identify the predictions with the lowest confidence and send them for manual labeling.

Finally, weakly supervised learning (Zhou, 2018; Zhang et al., 2022) can also reduce the amount of human labor to annotate the training samples. This approach is beneficial when there are large amounts of data, and manual labeling becomes infeasible (Roh et al., 2021). In weakly supervised learning, it is possible to automate the labeling process by defining a set of labeling functions (Ratner et al., 2017), hand-crafted rule-based classifiers. An example from the Snorkel tutorial[21] is a labeling function to sort emails as “spam”, using simply the presence of “http” in text metadata. A labeling function can leverage metadata or classifiers trained previously on similar tasks. Using multiple labeling functions helps to obtain a label score, which can be interpreted as a label probability, serving as weak supervision.

In conclusion, data annotation can be a time-consuming and expensive process, with many challenges (Rasmussen et al., 2022). However, quantity in some types of machine

[21] https://www.snorkel.org/use-cases/01-spam-tutorial

learning, such as deep neural networks, is a key to success. Therefore, much research is trying to alleviate this limitation by learning representations from unlabeled data, which is later discussed in Section 5.2.4.

5.2 Data Transformation

Once the data collection process has provided a set of initial raw data, different transformation techniques can be used to make the data suitable for a particular machine-learning model. This section presents a brief overview and discussion around distinct aspects of data transformation, which includes data integration or fusion, cleaning, reduction, representation, normalization or calibration, and augmentation.

Without considerable care, data leakage can be introduced during the transformation process for a competition. Before discussing the various types of transformations, we outline several ways information can be inadvertently leaked, urging dataset developers (and challenge organizers) to remain vigilant.

The ordering of observations (e.g., how the data is sorted) should not reveal any information about the targets. In our toy example of cats and dogs (mentioned in Section 4.1), it is evident that we would not want the images sorted such that all cat images precede the dog images. However, more subtle ordering issues can also cause leakage. An example occurred in the TalkingData AdTracking Fraud Detection Challenge[22] hosted on Kaggle. The data was sorted in such a way that if multiple events occurred within the same timestamp and any had a positive label, these were sorted below the negative labels. Although these occurrences were rare, they provided a small amount of leakage that some participants were able to exploit.

Another type of order-based leakage can be introduced while processing individual files. When preparing files for the competition, it is sometimes more convenient to process them by label, such as when opening images, removing metadata, and re-saving with a new observation ID (e.g., processing the folder of cats first and then the folder of dogs). However, if files are saved label-wise, participants can use the file timestamps to determine the label. A best practice is to process individual files randomly and reset the timestamps of the processed files after completion. Redundancy provides additional protection in case of a failure in one of the steps.

When creating a competition data processing pipeline, ensure that random sorting is repeatable by explicitly setting random seeds. Avoid using common random seeds (e.g., 0, 1, 123, 42), as these provide opportunities for participants to reverse-engineer the sorting method. Instead, use unique, difficult-to-guess random seeds that will not be reused in the future.

5.2.1 Integration and Fusion

Data integration or data fusion refers to the process of merging data or datasets from various sources into one dataset (Bleiholder and Naumann, 2008). For instance, if data are from the same relational database (e.g., SQL) but from different tables, then the JOIN operation[23] is a way to perform data fusion (which can expand sample and feature dimensions). In

[22] TalkingData AdTracking Fraud Detection Challenge: https://tinyurl.com/4c59vjky

[23] https://en.wikipedia.org/wiki/Join_(SQL)

this case, there often exists some matching identifier (e.g., an index that the tables can be joined on) that helps perform this task directly. Moreover, it is now possible to collect data from external databases (without matching identifiers), websites, or search engines in order to improve the predictive performance of machine learning models.

The type of algorithm used to perform data integration varies depending on the data type (e.g., tabular, image, sequence). During such operation, some typical problems to resolve are identification of matching entities (e.g., tabular), re-scaling (Section 5.2.3), spatial/temporal alignment or registration (e.g., 3D shapes, images, videos, Section 5.2.5), cleaning (e.g., removal of duplicates, imputation of missing values, Section 5.2.2), calibration and normalization (Section 5.2.5). Machine learning algorithms (Meng et al., 2020) can be used to address these problems and enable performing data integration or fusion. For example, the problem of 3D shape registration is usually resolved through the iterative closest point algorithm (Arun et al., 1987), based on the least-squares method. An advantage of the iterative closest point algorithm is that it iteratively estimates matching points, unlike the Procrustes-based method (Gower and Dijksterhuis, 2004). Another example is the Fuzzy-Join (Wang et al., 2011) literature, which intends to resolve the problem of inexact matching to merge two different datasets of tabular data.

It is important to mention that despite all the efforts, data integration is often far from perfect, and one should keep track of the source of each data as part of metadata to identify possible biases (e.g., the situation where all recorded patients with a given disease are coming from the same hospital).

5.2.2 Cleaning

Data cleaning refers to the process of improving the consistency of data (Ridzuan and Zainon, 2019). Some data records may be corrupted (e.g., download errors), incoherent (e.g., the same entity is represented by different tokens, or the same data has different associated labels), and may include missing data (e.g., partial information about a user). Data cleaning can be applied both on the input data (e.g., images, videos, text, etc.), the metadata, or any annotation (e.g., the target variable), jointly or separately. Processing all simultaneously is necessary to detect sampling bias with respect to individual samples or groups of samples and decide if the chosen data cleaning method is appropriate.

In fact, distinct methods have been proposed in the literature to deal with missing data, such as partial deletion, statistical imputation, interpolation, and Bayesian inference. However, missing data is problematic due to the risk of bias, which depends on the type of missing values, the relative size of the data that are missing, and the way of dealing with these missing values, which can have associated risks (e.g., yield false positives) and benefits (e.g., reduce false negatives) (Seijo-Pardo et al., 2019, 2018). For example, if missing data are missing at random, then they can be imputed (Van Buuren, 2018). However, if they are not missing at random, spectrum bias may be reinforced (e.g., increase bias toward already present patterns) with imputations. Similarly, samples with missing data could be removed (Guyon et al., 1996) but with the risk of introducing an exclusion bias.

Among data cleaning methods, Bayesian inference (Lew et al., 2021) is a family of methods offering good performance and automation capabilities. It can also leverage prior knowledge from domain expertise (i.e., understanding of the data), which is often the key to success.

5.2.3 Reduction

Data reduction corresponds to processes that reduce the information contained in data. It includes methods to reduce the dimensionality of feature space (often referred to as dimensionality reduction) and the number of samples (often referred to as sub-sampling). In some cases, data reduction can also refer to the re-scaling or re-sampling of signals according to spatial or temporal dimensions. Contrary to data augmentation and feature engineering, the goal of data reduction is to *reduce* spurious information in data, for instance, to accelerate learning or to make the predictor more robust by eliminating redundancy or noise in data. While several machine learning algorithms are naturally designed to separate important patterns from noise in the data, data-centric approaches such as data reduction can further facilitate this.

Concerning features, a canonical way to perform dimensionality reduction is the Principal Components Analysis (Wold et al., 1987) (PCA), which consists of building a subset of features, which are linear combinations of the original features and explains best the variance in data. PCA can be viewed as an ancestor of neural network auto-encoder methods for manifold learning. Indeed, Bourlard and Kamp (1988) have shown that for n inputs, a 2-layer network with $d < n$ hidden units, trained to reproduce its input on its output with mean-square-error, yields a representation projecting the data in the d directions of largest variance. Non-linear auto-encoders and their descendants (such as denoising auto-encoders (Bengio et al., 2013) and variational auto-encoders (Kingma and Welling, 2014)) provide a generalization of PCA. However, when working in a reduced space it can be required to reconstruct data in the original space. For example, in climate applications, the data are often extremely large and it is required to reduce the size of the data representation during learning but the prediction needs to be in the original space (Maulik et al., 2020). This step can be challenging with autoencoders; in many cases, reconstructed data do not satisfy validity constraints (i.e., data are not realistic). Some methods, such as grammar autoencoder (Kusner et al., 2017b) impose additional structure to alleviate this challenge.

Feature selection methods (Guyon and Elisseeff, 2003) are a particular case of dimensionality reduction methods that avoid replacing features with newly constructed features, which can facilitate explainability (i.e., interpretation about how a prediction is constructed from the inputs) in some applications. Like other methods of dimensionality reduction, removing redundancy and noise is the primary goal.

Also, more classical quantization can be performed to compress a signal. Quantization is the process of mapping a variable from an uncountable (e.g., real values) to a countable space. It is the core of lossy-compression algorithms and can be used to reduce the memory size of signals. A typical and fast way to perform quantization is through the k-means algorithm (Pollard, 1982).

Some dimensionality reduction methods are directly applicable to reduce the number of samples, such as PCA or clustering methods (Yen and Lee, 2009). It is also possible to leverage gradient-based methods to detect non-informative samples (Killamsetty et al., 2021). While sub-sampling can be used to balance the proportion of different classes in a dataset, it is unclear if it is consistently well-performing (García et al., 2012). The Imbalance-Learn Python package (Lemaître et al., 2017) provides a set of algorithms to perform sub-sampling.

The utility of data reduction is particularly important for spatiotemporal data, which quickly grow in size and face computational and memory limitations (Steadman et al., 2021). Although many of the previously introduced areas of data reduction have been extended to this setting, down-sampling of spatial resolution with bi-linear interpolation and strided sub-sampling over the temporal axis are often preferred in practice.

5.2.4 Representation

Data representation refers to a set of techniques that maps data to a numerical representation that is well-suited for the learning method. Basic data types can be real (e.g., height of a person in centimeters), discrete (e.g., number of users on a website), categorical nominal where there is no order on categories (e.g., type of vehicles such as car, scooter, or truck) and categorical ordinal where there is an order on categories without a clearly defined numerical scale (e.g., rating of a restaurant such as 'very bad', 'bad', 'medium', 'good' or 'very good'). In machine learning, data are generally represented as tensors (i.e., a n-dimensional matrix) even in the case of non-regular structures such as graphs (i.e., node-features, edges-features, connectivity which corresponds to $n = 3$). The problem of finding a good representation is key in machine learning (i.e., a representation that makes the model learn and generalize better).

In some cases, one wants to convert images into more high-level features. While techniques like deep neural networks can learn directly from the raw pixel values, there might be reasons to prefer more abstract and semantically richer, higher-level representations. A straightforward way of obtaining those is to use pre-trained models (Weiss et al., 2016) (benefiting from transfer-learning), such as I3D (Carreira and Zisserman, 2017) or R(2+1)D (Tran et al., 2018) for spatio-temporal feature representations. To do this, we can collect the output provided by the penultimate layer of a deep neural network to represent our new features. It is also possible to adapt the representation to our task. For example, we can perform a complete fine-tuning (i.e., all the weights of the neural network continue to be trained on our new task) or a simple linear-probing (i.e., all the weights are frozen we just change the last layer and train it). In particular, for many computer vision tasks, such a strategy provides a good enough initialization and speeds up the training on new datasets while being the most convenient strategy for small vision datasets.

Then, to overcome the difficulties of data annotation by reducing the quantity of annotated data, much research has focused on learning representations from unlabeled data. Learning representations from unlabeled data is now called self-supervised learning (Zhao et al., 2024) but directly corresponds to an extension of works previously classified under unsupervised learning. Self-supervised learning had many successes in natural language processing with methods such as Word2Vec (Mikolov et al., 2013), where a vector representation of words is learned, and arithmetic can be performed on such vectors having some plausible semantic interpretation. For instance, subtracting the vector representing "men" from that representing "king", then adding those of "women", yields a position in vector space close to "queen". The Word2Vec representation is obtained with a neural network, having at their input a part of the sentence, except for a missing central word, and at the output, the central word to be predicted (this algorithm is referred to as Continuous Bag of Words). Word2Vec has been a leap forward compared to previous bag-of-word representations, only based on frequencies of words in documents, such as TF-IDF (Sammut and Webb, 2010). Many other works followed the steps of Word2Vec and brought new achievements in the area of natural language processing (e.g., Glove (Pennington et al., 2014), fastText (Bojanowski et al., 2017), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), XLM-R (Conneau et al., 2020)). More recently, large language models have appeared (Brown et al., 2020) as the final realization of this methodology.

In computer vision, self-supervised learning is also applied to extract representations of images, which are later fine-tuned for supervised classification or regression tasks. In fact, using self-supervised learning on medical images has been shown to reduce the learning of spurious correlations (Goel et al., 2021) between input data and annotations, which also results in better performance (about +10% more on accuracy in some cases) (Azizi et al., 2021). The ideas from natural language processing and computer vision are now being generalized to other data representations such as graphs (e.g., Graph2Vec (Narayanan et al., 2017)) and spatiotemporal data.

All primordial methods of self-supervised learning are variants of auto-encoders, which learn a latent representation by first encoding and then decoding. They have recently been renamed “non-contrastive self-supervised learning methods”. A recent methodology for non-contrastive self-supervised learning is to use in-painting to learn to predict missing parts (i.e., occlusions) of an input image, where the occluded image is at the input of an auto-encoder and the missing part(s) at the output (Pathak et al., 2016). The limitation of non-contrastive self-supervised learning methods is that the model is not informed about counter examples, i.e., examples which are out-of-distribution or out of the support of the positive examples provided. This limitation motivated the need for contrastive learning, self-supervised learning (Chen et al., 2020), based on pairwise comparisons of similar and dissimilar examples. To that end, a Siamese neural network architecture (Bromley et al., 1993) is used, consisting of two identical networks whose outputs are compared with a contrastive loss function, such that agreement is maximized for similar or compatible inputs (e.g., two images of the same object, but from different views) and minimized in the case of dissimilarity or incompatibility (e.g., inputs represents different objects).

We illustrate the similarities and differences of self-supervised learning methods in Figure 5. Both have in common the mapping of the input data x to a new representation z (in blue). For non-contrastive (orange), often based on reconstruction schemes, the goal is to learn the representation z from x, which helps reconstruct the distorted data z' (e.g., jigsaw puzzle, in-painting). For contrastive (purple), the goal is to learn similar representations for similar entities and different representations for different entities. Similar entities are artificially simulated with data augmentation (e.g., x can be the image of a bird, and x' is the image of the same bird with a rotation), and then a contrastive loss can be used to enforce the contrastive idea.

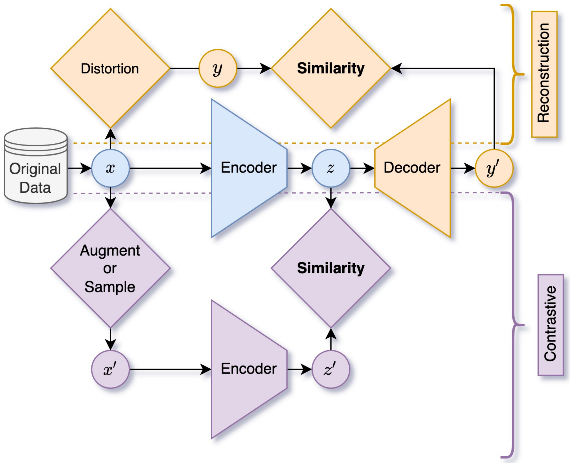

Figure 5: Self-Supervised Learning through Contrastive (purple) or Non-Contrastive (orange) Learning. The input data is x, and the representation learned is z for both.

5.2.5 Normalization and Calibration

Data normalization[24] or data calibration aims to get rid of some systematic bias, which may occur in data collection, due to several uncontrolled factors (e.g., a change in operator, a change in temperature, humidity, luminosity, amount of a certain reagent, etc.). Data calibration should not be confused with model calibration, which focuses on calibrating predictions to improve their probabilistic interpretability.

Linear normalizations or calibration simply amount to shifting and scaling features by an amount determined either by comparing samples to one another (e.g., normalization by dividing by the maximum), by the sample itself (e.g., normalization by diving by the norm of the sample) or by using some reference values (calibration). For example, standard feature normalization consists of removing the mean and dividing by the standard deviation feature-wise. This normalization is commonly performed on input data for neural networks, ensuring all features are unitless and spread over a similar range. In normalization, the quantities used to pre-process the data are directly estimated from these data. Therefore, one needs to be aware that the smaller the dataset is, the higher the variance of these estimates is (i.e., standard error is a function of the $\frac{1}{\sqrt{n}}$ where n is the number of examples), which can create unstable results when the dataset is small. In tabular data, normalizations are often carried out row-wise, column-wise, or both, depending on the nature of the application. Some practical way of performing normalization is through the Scikit-Learn[25] library which provides ready methods more or less sensitive to outliers such as: MinMaxScaler, MaxAbsScaler, StandardScaler, RobustScaler, Normalizer, QuantileTransformer and PowerTransformer.

Calibration can be thought of as a learning problem with a small training set. There are two types of calibrations, i.e., either making use of internal or external calibrants. An

[24] not to be confused with “database normalization”

[25] Scikit-Learn: https://tinyurl.com/bdfr7z62

internal calibrant is included in every sample. For instance, in chemistry, this would be a compound spiked in known quantity in a sample to adjust the scale of a titrating device; in photography, this would be e.g., landmarks positioned with a given geometry or a known pattern of given shapes and colors, captured together with the scene, serving as a reference to compensate for camera aberrations and adjust the color spectrum of pictures taken. In contrast, an external calibrant is a reference sample (with a well-defined pattern) inserted regularly in between regular samples—for instance, a chessboard image in photography, or the use of a water-solution in a spectrometer.

Internal calibrants are used when measurements constantly change, while external calibrants are suitable for slow drifts in recording conditions. In either case, the calibrants' ground truth (target values) are known. This allows users to train a simple predictive model (often just linear), which maps measured values to target values. The predictor can then interpolate between known values to correct the other measured values in the sample. This type of method is commonly used in computer vision for camera calibration (Zhang, 2000). Calibration is particularly needed for data fusion when samples are obtained from different sources, which relates this problem to data integration introduced in Section 5.2.1. Another example is the calibration of data from multiple sensors, such as in autonomous vehicles (Yeong et al., 2021).

5.2.6 Augmentation

Data augmentation is the process of artificially increasing the size of an already existing dataset (either with respect to samples or with respect to features (Liu et al., 2018)). It potentially enlarges the dataset by orders of magnitude at a reduced cost. It can be rule-based, such as in computer vision, where it is possible to perform changes in resolution, orientation, brightness, execute random crops/shifts, and include noise, among others (Shorten and Khoshgoftaar, 2019). It can also be learning-based, where the underlying distribution of the data is learned via generative models (e.g., generative adversarial networks (Goodfellow et al., 2020), variational auto-encoders (Kingma et al., 2016)), to then artificially sample from it. Although data augmentation can lead to overall improvements in performance (i.e., improved average metrics), it may introduce *selection bias* yielding an increase in performance for some classes or groups at the expense of others. This is related to the fact that designing dataset/task-specific regularizers without introducing selection bias remains an open research question (Balestriero et al., 2022). That is why proper model evaluation and selection needs to be conducted to quantify such effects. Some types of data augmentation are preferably executed during the training to avoid storing the augmented data. Finally, augmentation is often used jointly with other learning methods, especially self-supervised methods such as contrastive learning (Chen et al., 2020) and Siamese networks (Chen and He, 2021) to leverage the knowledge of proximity between original and augmented samples.

6 Evaluation

The goal of dataset evaluation is to assess whether a dataset meets its original dataset requirements, ensuring that it is suitable for training reliable and fair AI models. This includes verifying specified quality and quantity criteria, catching errors in dataset implementation, and identifying flaws in the dataset design. Our evaluation framework is based

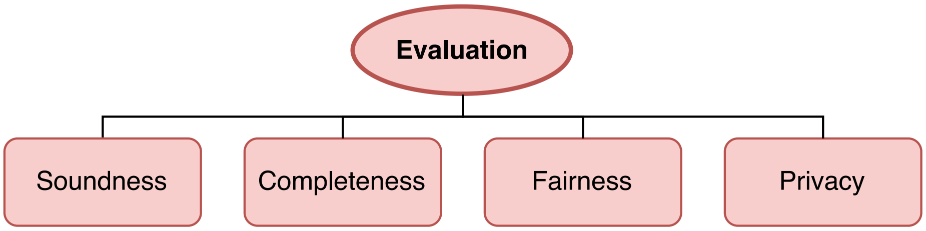

Figure 6: Categorization of sub-processes included in data evaluation.

on four key criteria: soundness, completeness, fairness, and privacy. Each of these criteria benefits from both qualitative (e.g., data visualization) and quantitative assessments (e.g., metrics).

6.1 Preliminaries: Inspection, Visualization, and Baselines

Visualization is key in evaluating a dataset (Figure 7). Data visualization tools should be prepared and used during data collection (Section 5.1) to help identify anomalies early. Common visualization techniques (Figure 7) include:

Heat or cluster maps: Useful for vectorial data to check for anomalous structures.

Pair plots: For datasets with few features, pair plots help visualize class separability. For many features, apply PCA first and use pair plots on the principal components.

Bar plots: Use bar plots to represent the frequency of examples in each class. These can help identify imbalances.

Commonly used library for data visualizations are: Matplotlib[26], Seaborn[27], and Microsoft SandDance[28].

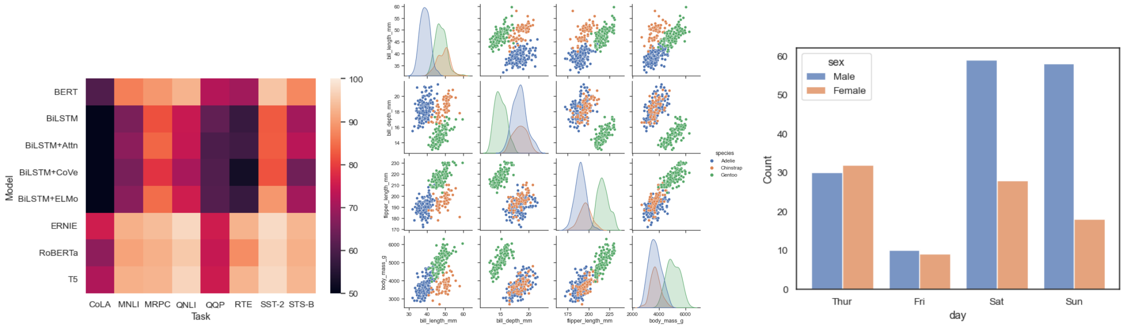

Figure 7: Example visualizations of (left) a heatmap, (middle) a pair-plot, and (right) class frequencies from the Seaborn library.

[26] https://matplotlib.org/

[27] https://seaborn.pydata.org/

[28] https://microsoft.github.io/SandDance/

It is recommended to apply baseline learning methods that tries to solve the task represented by the dataset early in the collection process. Baselines can help assess the difficulty of the dataset, and they can range from very basic to state-of-the-art. Also, the performance gap between the constant predictor (i.e., constant with respect to the input x, e.g., often the mode of target classes in the case of classification and the mean of targets in the case of regression) and state-of-the-art methods can be an indicator of learnable concepts in the data. Then, with trained baselines, other visualization can be done. For example, inspecting the confusion matrices for classification tasks is good practice. This can help determine "unfair" (Section 6.4) predictive performance across classes.

Visualization can be vital in revealing data leakage or bias, for example, by spotting illegitimate information in the input data (e.g., the case in which a red patch is present on all images of cancer cells while a green patch is present on all images of normal cells).

6.2 Soundness

A dataset is considered sound if it correctly results from its premises, which are the requirements, design, and implementation steps, and if these premises are themselves correct. For example, a design choice could be to identify cities by name and location (which would be correct), while another design could be to identify cities solely by name (which would be incorrect because names are not unique identifiers). Therefore, this will include verifying upstream steps and looking for inconsistencies, corruptions, and a good state of collected distributions. Classic unit tests can be implemented to partially check the soundness of the dataset. Next, we discuss some sources of consistency that should receive special attention.

Representation consistency relates to the problem of having a unique representation for the same entity across the dataset. For instance, when collecting articles from the press on the Internet, the same city can be written differently, such as "New York", "N.Y.", and "the Big Apple", which all represent the same city. Similarly, it is essential to check whether physical quantities are all measured using the same unit. In the case of tabular data and storage systems, improving representation consistency is often referred to as data deduplication (Xia et al., 2016).

Labelling consistency refers to self-agreement and inter-agreement among annotators, which is especially important when labeling was performed through crowd-sourcing as mentioned in Section 5.1.4. Suppose various annotators are participating in the annotation process. In that case, it is important to ensure they agree on the exact concept they are labeling and use the same definition and thresholds. Self-agreement is particularly useful in identifying low-quality annotators, while inter-agreement is suitable for estimating the task's difficulty. Krippendorff's Alpha-Reliability (Krippendorff, 2011) is an example of such a metric.

Outlier detection relates to identifying observations that appear inconsistent with other observations of the dataset (Hodge and Austin, 2004). Data visualization is particularly useful for detecting the presence of such samples. Other typical quantitative methods are based on the interquartile range (IQR, represented in box plots) and the Z-score. After identifying outliers, a domain expert can decide whether they result from errors.

Bias detection in the data generally concerns the identification of systematic outcomes of the dataset development process that results in a dataset that is not representative of the true observed phenomenon (Figure 2). For example, a lack of randomization of nuisance factors can result in spurious correlations. Collecting appropriate metadata (i.e., data necessarily not available for training but available for data evaluation) is essential to help detect such bias, including potential nuisance factors (e.g., temperature, humidity, luminosity, recording time, date, collection operator, etc.) and protected groups (e.g., age, gender, ethnicity, etc.) involved in societal bias (fairness, Section 6.4). Subjective bias (Section 6.4) coming from data annotation can also be mitigated by collecting proper metadata to reveal biases with respect to both the annotator and the data being annotated (Jacques Junior et al., 2021; Escalante et al., 2020). A machine learning model may then be trained using variables that are suspected causes of bias, either in isolation or in combinations. Feature selection methods are then applied to determine whether such variables are significantly predictive.

We now present use cases where the dataset was not sound.

Use Case: Spurious Correlation With Video Metadata in the "Looking at People" Challenge

The ChaLearn "Looking at People" Challenge on Self-Reported Personality Recognition (Palmero et al., 2022) adopted a dataset composed of large amounts of data (audio-visual, transcripts, metadata, etc.). However, one competitor team achieved promising results by analyzing the correlation between metadata and the self-assessment personality trait scores (the target variables) while proposing to use a random forest regressor trained solely on metadata features (i.e., age, gender, and number of sessions). That is, the competitor did not utilize the available 60 hours of provided audio-visual data and associated transcripts for training, which data creators believed to be crucial to addressing the problem. In other words, the data creators were not anticipating that a model trained on metadata features could accurately predict someone's personality, indicating that the adopted dataset may include unwanted bias. According to this competitor (and challenge results), a simple random forest regressor based on metadata features only should not be enough to outperform a method based on linguistic, audio-visual, and metadata features like the alternative model they evaluated in such a complex task as personality recognition.

Use Case: Spurious Correlation With File's Metadata in the "Whale" Challenge

A notable example of leakage from file metadata was a competition to detect whether underwater audio contained a Right whale up-call[29]*. A competitor scored an AUROC performance of 0.997 without actually reading the files (let alone making machine learning predictions). This was achieved simply by looking at the size-on-disk of the test files, the timestamp embedded in the audio clip filename, and the chronological order of the clips, which provided enough information to specify which files contained a whale up-call.*

[29] Kaggle's Right Whale Challenge: https://tinyurl.com/8d2uw69k

Use Case: Temporal Leakage in the "Predict Future Sales" Challenge

Another common soundness issue is time series leakage, wherein future data provided to competitors inadvertently reveals information. For instance, in the Corporación Favorita Grocery Sales Forecasting Kaggle competition, competitors were given oil prices up to the period of the test set[30]*. This might appear reasonable since the challenge was to predict store sales, not future oil prices. However, since the data originated from Ecuador, an oil-dependent country whose economic health is highly susceptible to oil price fluctuations, this inclusion constituted leakage. In real-world applications, predictive models would not have access to actual future oil prices, leading to inflated performance metrics in the challenge due to this leakage. Ideally, in time series competitions, competitors should only be provided with data up to the prediction horizon. Once their models generate predictions for the next period, they can receive the subsequent increment of data, continuing in this manner. While setting up a competition this way poses practical challenges, organizers might opt to provide the entire test data series upfront. Nevertheless, it is crucial to recognize that this approach introduces a form of leakage.*

6.3 Completeness

The completeness of a dataset is the attribute of a dataset to contain all required features describing sufficiently the problem at hand, as well as to properly select its samples (e.g., the number of i.i.d - independent and identically distributed - samples directly impacts the estimation of the mean estimate - standard error). Therefore, completeness can be evaluated with respect to samples or features but also with respect to metadata or hidden variables, which do not define the problem but have to be adequately sampled (e.g., randomized or factorial design) to avoid bias.

Feature-wise completeness is associated with the notion of causality, which defines that a cause can be necessary or sufficient. If X is a sufficient cause of Y, then the presence of X necessarily implies the subsequent occurrence of Y. However, another cause Z may alternatively cause Y. Thus, the presence of Y does not imply the prior occurrence of X. Then, if X is a necessary cause of Y, the presence of Y necessarily implies the prior occurrence of X. However, the presence of X does not imply that Y will occur. These relations matter to understanding the problem of confounding factors, which relates to a false association between variables such as X causing Y because of a third missing variable Z causing the two (Nunan et al., 2018).

Confounding variables can be divided into omitted variables and correlated noise (i.e., a spurious feature). As an example of an omitted variable: if X is "drinking coffee" and Y is "developing a lung cancer", some data can show that drinking coffee increases the risk factor of developing lung cancer. However, this is happening because a common cause Z, which is "smoking", was not taken into consideration and is associated with both "drinking coffee" and "developing lung cancer". An example of correlated noise is the background (e.g., road, sky, water) with the type of vehicle (e.g., car, plane,

[30] Corporación Favorita Grocery Sales Forecasting Kaggle competition: https://tinyurl.com/dttt4e86

boat). A picture of a car will rarely happen with a sky background; therefore, the background (the road) is predictive of the object's class (the car).

It can be tracked by identifying whether some common candidate confounding factors that are spuriously predictive of the target variable (individually or jointly). For sensor data, typical examples are temperature, humidity, luminosity, etc.; for survey data, typical examples are age, gender, ethnicity, etc. Classical feature importance methods can track how significant such associations are (Altmann et al., 2010). Proper metadata collection (also called protected attributes in fairness) can aid in this process (Bellamy et al., 2018). If some of these attributes cannot be recorded directly, then it is advisable to measure them indirectly. For example, suppose one suspects the image background could be a spurious feature. In that case, one may create mini-images containing only the background of the original images and try to predict the target variable from them (excluding the object of interest). If the model now predicts the object of interest, the background was utilized in the predictive model (Tian et al., 2018). If confounding bias is identified, the data collection process must be revised to alleviate it. Missing variables often cause excessive aleatoric uncertainty (or intrinsic randomness), which is the variability of the outcome given input variables because of unknown source factors. This variability needs to be assessed to make sure the prediction is performed within a reasonable confidence interval.

Sample-wise completeness is directly associated with the problem of selection bias, which can be divided into exclusion bias (removal of samples) and spectrum bias (only a subset of the target population was observed). Exclusion bias comes from a choice of the dataset development team. For example, it can result fram data cleaning, which may remove too many samples or may modify the data distribution by removing specific observations (e.g., creating an imbalanced dataset). It can also be a choice of filter in a search engine or a selected window of observation. A well-known bias in an academic dataset is the recruitment of graduate students as subjects. Testing sample-wise completeness depends on the assumption made on the problem. In the i.i.d. samples case, it can be tested through the classic generalization error using learning curves techniques (Mohr and van Rijn, 2022, 2023).

6.4 Fairness

Fairness has recently attracted attention in the machine learning community (Bird et al., 2019; Mehrabi et al., 2021) after several flaws arising from misuse of biased data being reported by the media such as the Guardian[31] or the Washington Post[32]. In the context of decision-making, and according to (Mehrabi et al., 2021), fairness is the absence of any prejudice or favoritism toward an individual or group based on their inherent or acquired characteristics. Therefore, an unfair algorithm is one whose decisions are skewed toward a particular group of people.

Machine learning algorithms can inherently transfer bias from data to model. When black-box models are used (e.g., most deep learning architectures), results are usually diffi-

[31] The Guardian, 2022: http://tinyurl.com/yfeeu2sx

[32] The Washington Post, 2020: http://tinyurl.com/4kv4nf5v

cult to explain and interpret, making bias mitigation even harder. Hence, a biased dataset used for training or evaluating machine learning methods can negatively impact outcomes. For instance, a face recognition method trained on male faces may not generalize well to female faces. To mitigate such problems, in most countries, the law protects against a discriminatory decision (e.g., about hiring or condemning) based on protected attributes, which include gender, age, and ethnicity, among others (Barocas et al., 2023). One way of ensuring some level of fairness at the data collection level is to ensure that samples are group-balanced, i.e., that there is an approximately equal number of samples in all groups resulting from combinations of protected attributes and labels. Of course, this may be impractical, and it may be more feasible to record the protected attribute to later correct bias at the machine learning level, with in-processing (Wan et al., 2022) (i.e., during learning, which is categorized into explicit and implicit, where the former directly incorporates fairness metrics in training objectives, and the latter focuses on refining latent representation learning) or post-processing techniques (i.e., after learning). However, recording protected variables may raise privacy issues (Section 6.5).

Data annotation may also be a source of social bias. The machine learning and computer vision communities are paying more attention to this problem as it relates to fairness (Bird et al., 2019). Recent work reports different types of subjective biases coming from crowdsourced annotations. Although biases produced by human perception have been widely studied in sociology and psychology (e.g., gender (Oh et al., 2019) or attractiveness (Talamas et al., 2016) bias), little attention has been given to subjective bias analysis (Shen et al., 2019; Quadrianto et al., 2019; Robinson et al., 2020; Yan et al., 2020) beyond the perspective of explainable models (Escalante et al., 2017; Huk Park et al., 2018; Pérez Principi et al., 2019; Escalante et al., 2020). Moreover, as perception depends on the observer, the relationship between annotators and the entity being annotated could explain how some perception biases are produced, which is an almost unexplored area in computer vision and machine learning. However, this would require a dedicated discussion around privacy and ethical issues (Jacques Junior et al., 2021).

Over the past few years, a vast number of scientific events and studies appeared intending to stimulate discussion and advance the state of the art on fairness and bias mitigation methods (e.g., ACM FAccT[33]), explainability and interpretability (e.g., Escalante et al. (2017); Huk Park et al. (2018)). Distinct definitions and metrics for fairness have been proposed and discussed, like “fairness through unawareness”, “individual fairness”, “demographic parity”, “equalized odds”, “equality of opportunity” or “counterfactual fairness” (Kusner et al., 2017a; Ashokan and Haas, 2021; Bellamy et al., 2018). Although there is no standard definition of fairness that could be used for all types of problems, researchers must be attentive to possible fairness issues and consult with social scientists and ethics specialists as needed. For instance, it is advisable to have data collection protocols reviewed by an Institutional Review Board (see Section 3) even though this does not guarantee to solve all possible problems.

[33] https://facctconference.org

6.5 Privacy

When a dataset contains human-related data, privacy and data protection become mandatory and will impact all aspects of the dataset development. Data protection regulations (e.g., the European General Data Protection Regulation, GDPR (Voigt and Bussche, 2017)) define different levels of protection depending on each type of data, and each level has different requirements for processing and storing. Moreover, the classification of personal data differs in each regulation and can change regularly. Human-related data require collecting an explicit consent form before storing any data, with a clear and understandable description of the collected data, how it will be used, who will have access to it, and how long it will be stored.

Data are considered anonymized if there is no possibility to identify a subject using the provided data. In cases where data curators maintain a correspondence between published data and originally captured data, we cannot consider the data anonymized, as it is possible to recover a subject's identity. In this case, the data is pseudo-anonymized, where without the link between real identity and published data, no one can recover the real identity of a subject in the dataset. Pseudo-anonymized data allow data curators to remove data from an individual if necessary, but they pose a security risk since if this link is compromised, real identities can be recovered. Avoiding data leakage is crucial to maintaining privacy, as it prevents unauthorized access to sensitive information and ensures compliance with legal and ethical standards.

In the context of anonymization and pseudo-anonymization, k-anonymity is an important measure. The measure of k-anonymity implies that given one entry of the datasets (i.e., information of a specific individual), it contains at least $k-1$ identical entries in the dataset corresponding to other individuals. The minimum recommended value for k is three, although larger values ensure better anonymization. For many public datasets, re-identification is easy even when data seems anonymous. For example, Sweeney (2000) demonstrates that 86% of the U.S. population can be uniquely identified with just three "quasi-identifiers": zip code, gender, and date of birth. Although there are easy ways to increase the k-value, e.g., by binning variables, if a dataset contains sensitive data, it is a good idea to apply one of the many existing anonymization algorithms (Casas-Roma et al., 2012). Finally, it is essential to ensure that the anonymization process does not affect the usefulness of the dataset. Carmona et al. (2019) provide an introduction to the subject regarding health data.

Differential privacy (DP) (Dwork, 2008) and the possibility of replacing real-world data with realistic synthetic data providing some privacy guarantees have gained some recent attention. Sablayrolles et al. (2020) propose an apparatus in which an ideal attacker having maximum information evaluates whether such synthetic data are protected against membership inference attacks (i.e., determining whether a sample was or not part of the data used to train the generative model).

7 Distribution and Maintenance

Dataset distribution is about making the developed dataset accessible to others, while maintenance are all tasks and processes required to maintain the dataset accessibility and to perform changes on the dataset or how it is distributed to improve its accessibility or quality. Those two tasks should often be considered together. Depending on the structure of the dataset (e.g., size, data type, format) those can be simple or complex tasks. Taking into account the requirements of the dataset (Section 3), one may select the maintenance and distribution technologies/strategies that are cost-effective, sustainable, and supportable with available resources (economic and human). Next, we present the aspects we consider the most relevant to be considered:

Ownership and licensing: The distribution of a dataset includes essential legal requirements. The dataset creators have to define the usage and responsibility of distributed data. To this end, the dataset can be distributed under copyrights, specific licensing (Benjamin et al., 2019) (e.g., open-source[34], creative commons[35]) or terms of use (ToU). Licensing not only includes the dataset creators but also the object of shared data, such as information about individuals, whether the information is personal or medical, whether the individual agreed to distribute this data or whether regulation exists for this type of data (e.g., General Data Protection Regulation[36] in Europe, California Privacy Rights Act[37]). Therefore, the maintenance plan may include all the processes required by the data protection regulations, such as the capacity to remove all the data for an individual in case it is required or the partial or total elimination of the dataset and/or original data. The distribution tools may also support such actions.

Hosting platforms: The dataset can be hosted in a Cloud service (e.g., Microsoft Azure, Google Cloud, or Amazon Web Services) and benefit from optimized downloading services if the source (where the dataset is stored) and the target (where the dataset is downloaded) platforms are from the same service (e.g., Google Drive and Google Colab with `gdown`). It is also possible to directly store the data on a web server so that it can be downloaded utilizing the `http` protocol, the File Transfer Protocol (FTP), and the Secure Copy Protocol (SCP). More recent open-science services such as Globus[38] can also be set up to optimize the transfer of data.

Evolution and versioning: Datasets often evolve in time to fix errors or include new data or labels. Also, flaws inside the dataset can be discovered later through active usage (Wang et al., 2022). A good example is linked to the evolution of machine learning research. Recently, privacy preservation and fairness in machine learning have become a priority. Therefore, the ImageNet dataset was updated to anonymize individuals appearing in pictures or filter out problematic samples[39]. Users were informed about these changes through ImageNet's website.

Due to these continuous changes, it is essential to attribute a version or unique identifier to the dataset to differentiate it in case of modification. Open-science (free of

[34] https://opensource.org/licenses

[35] https://creativecommons.org/

[36] https://gdpr.eu/what-is-gdpr

[37] California Privacy Rights Act: https://tinyurl.com/mr38ctvu

[38] https://www.globus.org/

[39] ImageNet: https://tinyurl.com/54ux9bsu

access) websites such as Zenodo[40] and arXiv[41] now provide digital object identifiers (DOI) which can be used for this purpose. Sometimes it is also possible to use data version control[42] if the data is not required to be completely removed (e.g., for scientific reproducibility) so that versions of the data can be tracked. Data version control is essential when having an evolving dataset to keep track of the versions and give the possibility to trace which dataset was used to train a particular model.

Data format: When distributing data, it is often important to compile it under a compressed format so that it can be downloaded faster. Such format can be `.tar`, `.zip`, and `.gz` to cite a few. It is important to inform users about the decompressed size when using a compressed format. Although the format can change due to the evolution of the dataset, it is desirable to maintain backward compatibility as much as possible. This backward compatibility is a general software principle that is also valid for datasets. A significant format change can limit its usability.

Dissemination: Lastly, in the case of a public dataset, it is essential to communicate its existence. Organizing related competitions and events in international conferences can be an excellent opportunity to present the dataset and a first benchmark using it. The NeurIPS conference promoted a specialized track to publish new datasets and benchmarks that can help showcase such datasets. More details are provided by Richard et al. (2024).

8 Conclusion

Datasets are an essential aspect of scientific benchmarks and competitions for machine learning. More importantly, properly designed and evaluated datasets are extremely important for developing trustworthy and robust artificial intelligence systems. In this chapter, we aimed to specify the dataset development process. We have categorized the various steps that should be undertaken in the dataset development cycle, i.e., documentation, requirements, design, implementation, evaluation, and distribution and maintenance. We approached dataset development as an agile process that is often iterative and requires interactions between its sub-processes. However, we acknowledge that every dataset development process can be different, and there is the possibility to emphasize or skip certain parts of this process. For example, in some cases, emphasis is placed on the design phase, whereas in other cases, the maintenance phase is limited (e.g., when there is no ability to improve the dataset after it has been released).

While we have attempted to give a broad overview of the dataset development process, this is by no means exhaustive. When developing a dataset, one must take care to not introduce any bias. Every dataset development process can introduce its distinct type of bias. Furthermore, while this chapter focuses on the dataset development process, many machine learning benchmarks and competitions also include a stage that evaluates the models trained on these data. Typically, this involves splitting the dataset into a train and test set, which, when not appropriately addressed, can induce other types of bias (e.g.,

[40] https://zenodo.org/

[41] https://arxiv.org/

[42] https://dvc.org

information bias and data leakage). While we briefly touched upon the concept of data leakage and how to consider this pro-actively in the dataset development cycle, how to avoid this completely in the concept of competitions is out of the scope of this chapter.

With this chapter, we aimed to harmonize some terminologies from the dataset development process as well as bring together several directions of the literature that we expect to be taken into consideration when developing new datasets.

9 Acknowledgment

This material is based upon work partially supported by the U.S. Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research, under Contract DE-AC02-06CH11357. This material is based upon work partially supported by the ANR Chair of Artificial Intelligence HUMANIA ANR-19-CHIA-0022 and TAILOR EU Horizon 2020 grant 952215. This work has been partially supported by the Spanish projects PID2022-136436NB-I00 and PID2022-138721NB-I00, and by ICREA under the ICREA Academia program.

References

Eirikur Agustsson, Radu Timofte, Sergio Escalera, Xavier Baró, Isabelle Guyon, and Rasmus Rothe. Apparent and real age estimation in still images with deep residual regressors on appa-real database. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 87–94, 2017.

Ashly Ajith and G. Gopakumar. Domain adaptation: A survey. In *Computer Vision and Machine Intelligence - Lecture Notes in Networks and Systems*, pages 591–602. Springer, Singapore, 2023.

André Altmann, Laura Toloşi, Oliver Sander, and Thomas Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 2010.

K. S. Arun, Thomas S. Huang, and Steven D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 9(5):698–700, 1987.

Ashwathy Ashokan and Christian Haas. Fairness metrics and bias mitigation strategies for rating predictions. *Information Processing and Management*, 58(5):102646, 2021.

Shekoofeh Azizi, Basil Mustafa, Fiona Ryan, Zachary Beaver, Jan Freyberg, Jonathan Deaton, Aaron Loh, Alan Karthikesalingam, Simon Kornblith, Ting Chen, et al. Big self-supervised models advance medical image classification. In *IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 3478–3488, 2021.

Randall Balestriero, Leon Bottou, and Yann LeCun. The effects of regularization and data augmentation are class dependent. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022*, pages 37878–37891, 2022.

Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. MIT Press, 2023.

Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *CoRR*, abs/1810.01943, 2018.

Emily M. Bender and Batya Friedman. Data Statements for Natural Language Processing: Toward Mitigating System Bias and Enabling Better Science. *Transactions of the Association for Computational Linguistics*, 6:587–604, 12 2018.

Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising autoencoders as generative models. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013*, pages 899–907, 2013.

Misha Benjamin, Paul Gagnon, Negar Rostamzadeh, Christopher Joseph Pal, Yoshua Bengio, and Alex Shee. Towards standardization of data licenses: The Montreal data license. *CoRR*, abs/1903.12262, 2019.

Karan Bhanot, Miao Qi, John S Erickson, Isabelle Guyon, and Kristin P Bennett. The problem of fairness in synthetic healthcare data. *Entropy*, 23(9):1165, 2021.

Karan Bhanot, Ioana Baldini, Dennis Wei, Jiaming Zeng, and Kristin P. Bennett. Downstream fairness caveats with synthetic healthcare data. *CoRR*, abs/2203.04462, 2022.

Sarah Bird, Krishnaram Kenthapadi, Emre Kiciman, and Margaret Mitchell. Fairness-aware machine learning: Practical challenges and lessons learned. In *International Conference on Web Search and Data Mining*, pages 834–835, 2019.

Bernd Bischl, Giuseppe Casalicchio, Matthias Feurer, Pieter Gijsbers, Frank Hutter, Michel Lang, Rafael Gomes Mantovani, Jan N. van Rijn, and Joaquin Vanschoren. OpenML benchmarking suites. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Jens Bleiholder and Felix Naumann. Data fusion. *ACM computing surveys (CSUR)*, 41(1): 1–41, 2008.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

Hervé Bourlard and Yves Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4):291–294, 1988.

Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In *Advanced Lectures on Machine Learning, ML Summer Schools 2003*, volume 3176 of *Lecture Notes in Computer Science*, pages 169–207. Springer, 2003.

Alexander Braylan, Omar Alonso, and Matthew Lease. Measuring annotator agreement generally across complex structured, multi-object, and free-text annotation tasks. In *Proceedings of the ACM Web Conference*, pages 1720–1730, 2022.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "Siamese" time delay neural network. In *Advances in Neural Information Processing Systems 6, [7th NIPS Conference]*, pages 737–744, 1993.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, pages 1877–1901, 2020.

Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

Guilherme Camargo, Pedro H. Bugatti, and Priscila T. M. Saito. Active semi-supervised learning for biological data classification. *PLOS ONE*, 15(8):1–20, 2020.

Fer Carmona, Jordi Conesa, and Jordi Casas-Roma. Towards the analysis of how anonymization affects usefulness of health data in the context of machine learning. In *International Symposium on Computer-Based Medical Systems*, pages 604–608, 2019.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017.

Jordi Casas-Roma, Jordi Herrera-Joancomartí, and Vicenç Torra. Comparing random-based and k-anonymity-based algorithms for graph anonymization. In *Modeling Decisions for Artificial Intelligence*, volume 7647, pages 197–209. Springer, 2012.

Alessandro Checco, Kevin Roitero, Eddy Maddalena, Stefano Mizzaro, and Gianluca Demartini. Let's agree to disagree: Fixing agreement measures for crowdsourcing. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 5(1):11–20, 2017.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International conference on machine learning, ICML 2020*, Proceedings of Machine Learning Research (PMLR), pages 1597–1607, 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15745–15753, 2021.

Gary Chin. *Agile Project Management: How to Succeed in the Face of Changing Project Requirements.* Amacom, 2004. ISBN 9780814427361.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 8440–8451. Association for Computational Linguistics, 2020.

Ciprian Adrian Corneanu, Marc Oliu Simón, Jeffrey F. Cohn, and Sergio Escalera Guerrero. Survey on RGB, 3D, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(8):1548–1568, 2016.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,NAACL-HLT 2019*, pages 4171–4186. Association for Computational Linguistics, 2019.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

Cynthia Dwork. Differential privacy: A survey of results. In *Theory and Applications of Models of Computation*, pages 1–19. Springer Berlin Heidelberg, 2008.

Maurizio Di Paolo Emilio. *Data Acquisition Systems: From Fundamentals to Applied Design.* Springer, 2013. ISBN 1461442133.

Hugo Jair Escalante, Isabelle Guyon, Sergio Escalera, Júlio C. S. Jacques Júnior, Meysam Madadi, Xavier Baró, Stéphane Ayache, Evelyne Viegas, Yagmur Güçlütürk, Umut Güçlü, Marcel A. J. van Gerven, and Rob van Lier. Design of an explainable machine learning challenge for video interviews. In *International Joint Conference on Neural Networks (IJCNN)*, pages 3688–3695, 2017.

Hugo Jair Escalante, Heysem Kaya, Albert Ali Salah, Sergio Escalera, Yağmur Güçlütürk, Umut Güçlü, Xavier Baró, Isabelle Guyon, Julio C. S. Jacques, Meysam Madadi, Stephane Ayache, Evelyne Viegas, Furkan Gurpinar, Achmadnoer Sukma Wicaksana, Cynthia Liem, Marcel A. J. Van Gerven, and Rob Van Lier. Modeling, recognizing, and explaining apparent personality from videos. *IEEE Transactions on Affective Computing*, 13(2):894–911, 2020.

Vicente García, José Salvador Sánchez, and Ramón Alberto Mollineda. On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. *Knowledge-Based Systems*, 25(1):13–21, 2012.

Hector Garcia-Molina, Manas Joglekar, Adam Marcus, Aditya Parameswaran, and Vasilis Verroios. Challenges in data crowdsourcing. *IEEE Transactions on Knowledge and Data Engineering*, 28(4):901–911, 2016.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Commununications of the ACM*, 64(12):86–92, 2021.

Karan Goel, Albert Gu, Yixuan Li, and Christopher Ré. Model patching: Closing the subgroup performance gap with data augmentation. In *9th International Conference on Learning Representations, ICLR 2021*. OpenReview.net, 2021.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

John C Gower and Garmt B Dijksterhuis. *Procrustes Problems*. Oxford University Press, 2004. ISBN 9780198510581.

Robert M. Gray and David L. Neuhoff. Quantization. *IEEE Transactions on Information Theory*, 44(6):2325–2383, 1998.

Yingjie Gu, Zhong Jin, and Steve C. Chiu. Combining active learning and semi-supervised learning using local and global consistency. In *Neural Information Processing*, volume 8834 of *Lecture Notes in Computer Science*, pages 215–222. Springer, 2014.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

Isabelle Guyon, Nada Matic, and Vladimir Vapnik. *Discovering informative patterns and data cleaning*, pages 181–203. American Association for Artificial Intelligence, USA, 1996.

Isabelle Guyon, John Makhoul, Richard Schwartz, and Vladimir Vapnik. What size test set gives good error rate estimates? *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 20(01):52–64, 1998.

David Harrison Jr and Daniel L Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management*, 5(1):81–102, 1978.

John Rowland Higgins. *Sampling theory in Fourier and signal analysis: foundations*. Oxford Science Publications, 1996. ISBN 0198596995.

Victoria Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial intelligence review*, 22(2):85–126, 2004.

Sarah Holland, Ahmed Hosny, Sarah Newman, Joshua Joseph, and Kasia Chmielinski. The dataset nutrition label: A framework to drive higher data quality standards. *CoRR*, abs/1805.03677, 2018.

Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Multimodal explanations: Justifying decisions and pointing to the evidence. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8779–8788, 2018.

Ben Hutchinson, Andrew Smart, Alex Hanna, Emily Denton, Christina Greer, Oddur Kjartansson, Parker Barnes, and Margaret Mitchell. Towards accountability for machine learning datasets: Practices from software engineering and infrastructure. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 560–575, 2021.

Julio C. S. Jacques Junior, Agata Lapedriza, Cristina Palmero, Xavier Baro, and Sergio Escalera. Person perception biases exposed: Revisiting the first impressions dataset. In *IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pages 13–21, 2021.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Shachar Kaufman, Saharon Rosset, Claudia Perlich, and Ori Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):1–21, 2012.

Krishnateja Killamsetty, S Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Gradmatch: Gradient matching based data subset selection for efficient deep model training. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, Proceedings of Machine Learning Research (PMLR), pages 5464–5474, 2021.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016*, pages 4743–4751, 2016.

Bernard Koch, Emily Denton, Alex Hanna, and Jacob G. Foster. Reduced, reused and recycled: The life of a dataset in machine learning research. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021*, 2021.

Klaus Krippendorff. Computing krippendorff's alpha-reliability. Technical report, Annenberg School for Computing, 2011.

Aditya Kuppa, Lamine Aouad, and Nhien-An Le-Khac. Towards improving privacy of synthetic datasets. In *Privacy Technologies and Policy*, pages 106–119, Cham, 2021. Springer International Publishing. ISBN 978-3-030-76663-4.

Matt J. Kusner, Joshua R. Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 4066–4076, 2017a.

Matt J. Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, volume 70 of *Proceedings of Machine Learning Research (PMLR)*, pages 1945–1954, 2017b.

Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.

Alexander Lew, Monica Agrawal, David Sontag, and Vikash Mansinghka. Pclean: Bayesian data cleaning at scale with domain-specific probabilistic programming. In *International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research (PMLR)*, pages 1927–1935, 2021.

Bo Liu, Xudong Wang, Mandar Dixit, Roland Kwitt, and Nuno Vasconcelos. Feature space transfer for data augmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9090–9098, 2018.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *CoRR*, abs/1907.11692, 2019.

Javier Maroto and Antonio Ortega. Efficient worker assignment in crowdsourced data labeling using graph signal processing. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2271–2275, 2018.

Romit Maulik, Romain Egele, Bethany Lusch, and Prasanna Balaprakash. Recurrent neural network architecture search for geophysical emulation. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press, 2020.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys*, 54(6):1–35, 2021.

Tong Meng, Xuyang Jing, Zheng Yan, and Witold Pedrycz. A survey on machine learning for data fusion. *Information Fusion*, 57:115–129, 2020.

Milagros Miceli, Tianling Yang, Laurens Naudts, Martin Schuessler, Diana Serbanescu, and Alex Hanna. Documenting computer vision datasets: An invitation to reflexive data practices. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 161–172, 2021.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013*, 2013.

Felix Mohr and Jan N. van Rijn. Learning curves for decision making in supervised machine learning - A survey. *CoRR*, abs/2201.12150, 2022.

Felix Mohr and Jan N. van Rijn. Fast and informative model selection using learning curve cross-validation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45(8):9669–9680, 2023.

Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. Graph2vec: Learning distributed representations of graphs. *CoRR*, abs/1707.05005, 2017.

Sergey Nikolenko. *Synthetic Data for Deep Learning*. Springer, Cham, 2021. ISBN 978-3-030-75177-7. Part of the Springer Optimization and Its Applications book series (SOIA, volume 174).

Stefanie Nowak and Stefan Rüger. How reliable are annotations via crowdsourcing: A study about inter-annotator agreement for multi-label image annotation. In *Proceedings of the International Conference on Multimedia Information Retrieval*, page 557–566, 2010.

David Nunan, Jeffrey Aronson, and Clare Bankhead. Catalogue of bias: attrition bias. *BMJ Evidence-Based Medicine*, 23(1):21–22, 2018.

DongWon Oh, Elinor A. Buck, and Alexander Todorov. Revealing hidden gender biases in competence impressions of faces. *Psychological Science*, 30(1):65–79, 2019.

Cristina Palmero, German Barquero, Julio C. S. Jacques Junior, Albert Clapés, Johnny Núñez, David Curto, Sorina Smeureanu, Javier Selva, Zejian Zhang, David Saeteros, David Gallardo-Pujol, Georgina Guilera, David Leiva, Feng Han, Xiaoxue Feng, Jennifer He, Wei-Wei Tu, Thomas B. Moeslund, Isabelle Guyon, and Sergio Escalera. Chalearn LAP challenges on self-reported personality recognition and non-verbal behavior forecasting during social dyadic interactions: Dataset, design, and results. In *Understanding Social Behavior in Dyadic and Small Group Interactions*, volume 173 of *Proceedings of Machine Learning Research (PMLR)*, pages 4–52, 2022.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544, 2016.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

Ricardo Darío Pérez Principi, Cristina Palmero, Júlio C. S. Jacques Júnior, and Sergio Escalera. On the effect of observed subject biases in apparent personality analysis from audio-visual signals. *IEEE Transactions on Affective Computing*, pages 1–14, 2019.

David Pollard. Quantization and the method of k-means. *IEEE Transactions on Information theory*, 28(2):199–205, 1982.

Víctor Ponce-López, Baiyu Chen, Marc Oliu, Ciprian A. Corneanu, Albert Clapés, Isabelle Guyon, Xavier Baró, Hugo Jair Escalante, and Sergio Escalera. ChaLearn LAP 2016: First round challenge on first impressions - dataset and results. In *European Conference on Computer Vision Workshop (ECCVW)*, pages 400–418, 2016.

Novi Quadrianto, Viktoriia Sharmanska, and Oliver Thomas. Discovering fair representations in the data domain. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8219–8228, 2019.

Rossana Queiroz, Marcelo Cohen, Juliano L. Moreira, Adriana Braun, Julio C. Jacques Junior, and Soraia Raupp Musse. Generating facial ground truth with synthetic faces. In *Conference on Graphics, Patterns and Images*, pages 25–31, 2010.

Christoffer Bøgelund Rasmussen, Kristian Kirk, and Thomas B. Moeslund. The challenge of data annotation in deep learning - a case study on whole plant corn silage. *Sensors*, 22(4), 2022.

Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, 11(3):269–282, 2017.

Magali Richard, Yuna Blum, Justin Guinney, Gustavo Stolovitzky, and Adrien Pavão. AI competitions and benchmarks, practical issues: Proposals, grant money, sponsors, prizes, dissemination, publicity. *CoRR*, abs/2401.04452, 2024.

Fakhitah Ridzuan and Wan Mohd Nazmee Wan Zainon. A review on data cleansing methods for big data. *Procedia Computer Science*, 161:731–738, 2019.

Joseph P. Robinson, Gennady Livitz, Yann Henon, Can Qin, Yun Fu, and Samson Timoner. Face recognition: Too bias, or not too bias? In *Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1–10, 2020.

Yuji Roh, Geon Heo, and Steven Euijong Whang. A survey on data collection for machine learning: A big data - ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1328–1347, 2021.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Herve Jegou. Radioactive data: tracing through training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119 of *Proceedings of Machine Learning Research (PMLR)*, pages 8326–8335, 2020.

Claude Sammut and Geoffrey I. Webb, editors. *TF–IDF*, pages 986–987. Springer, Boston, MA, 2010. ISBN 978-0-387-30164-8.

Nabeel Seedat, Fergus Imrie, and Mihaela van der Schaar. DC-check: A data-centric AI checklist to guide the development of reliable machine learning systems. *CoRR*, abs/2211.05764, 2022.

Borja Seijo-Pardo, Amparo Alonso-Betanzos, Kristin P. Bennett, Verónica Bolón-Canedo, Isabelle Guyon, Julie Josse, and Mehreen Saeed. Analysis of imputation bias for feature selection with missing data. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 655–660, 2018.

Borja Seijo-Pardo, Amparo Alonso-Betanzos, Kristin P. Bennett, Verónica Bolón-Canedo, Julie Josse, Mehreen Saeed, and Isabelle Guyon. Biases in feature selection with missing data. *Neurocomputing*, 342:97–112, 2019.

Burr Settles. Active learning literature survey. *Computer Sciences Technical Report, 1648. University of Wisconsin-Madison, Department of Computer Sciences*, 2009.

Judy Hanwen Shen, Agata Lapedriza, and Rosalind W. Picard. Unintentional affective priming during labeling may bias labels. In *International Conference on Affective Computing and Intelligent Interaction*, pages 587–593, 2019.

Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.

Liam Steadman, Nathan Griffiths, Stephen Jarvis, Mark Bell, Shaun Helman, and Caroline Wallbank. Kd-str: A method for spatio-temporal data reduction and modelling. *ACM/IMS Transactions on Data Science*, 2(3), 2021.

Latanya Sweeney. Simple demographics often identify people uniquely. *Carnegie Mellon University, Data Privacy*, 2000.

Sean N. Talamas, Kenneth I. Mavor, and David I. Perrett. Blinded by beauty: Attractiveness bias and accurate perceptions of academic performance. *PLOS ONE*, 11(2):1–18, 2016.

Xiu Tang, Sai Wu, Gang Chen, Ke Chen, and Lidan Shou. Learning to label with active learning and reinforcement learning. In *Database Systems for Advanced Applications*, pages 549–557. Springer International Publishing, 2021. ISBN 978-3-030-73197-7.

Maoqing Tian, Shuai Yi, Hongsheng Li, Shihua Li, Xuesen Zhang, Jianping Shi, Junjie Yan, and Xiaogang Wang. Eliminating background-bias for robust person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5794–5803, 2018.

Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(11):1958–1970, 2008.

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6450–6459, 2018.

Ihsan Ullah, Dustin Carrión-Ojeda, Sergio Escalera, Isabelle Guyon, Mike Huisman, Felix Mohr, Jan N. van Rijn, Haozhe Sun, Joaquin Vanschoren, and Phan Anh Vu. Meta-album: Multi-domain meta-dataset for few-shot image classification. In *Advances in*

Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, 2022.

Stef Van Buuren. *Flexible imputation of missing data.* CRC press, 2018.

Jesper E. van Engelen and Holger H. Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020.

Joaquin Vanschoren, Jan N Van Rijn, Bernd Bischl, and Luis Torgo. OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014.

Paul Voigt and Axel von dem Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide.* Springer, 2017.

Mingyang Wan, Daochen Zha, Ninghao Liu, and Na Zou. In-processing modeling techniques for machine learning fairness: A survey. *ACM Transactions on Knowledge Discovery from Data*, 2022.

Angelina Wang, Alexander Liu, Ryan Zhang, Anat Kleiman, Leslie Kim, Dora Zhao, Iroha Shirai, Arvind Narayanan, and Olga Russakovsky. Revise: A tool for measuring and mitigating bias in visual datasets. *International Journal of Computer Vision*, 130(7): 1790–1810, 2022.

Jiannan Wang, Guoliang Li, and Jianhua Fe. Fast-join: An efficient method for fuzzy token matching based string similarity join. In *IEEE International Conference on Data Engineering*, pages 458–469, 2011.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

Mark D. Wilkinson et al. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3(1):160018, 2016.

Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

Erroll Wood, Tadas Baltrusaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J. Cashman, and Jamie Shotton. Fake it till you make it: face analysis in the wild using synthetic data alone. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021*, pages 3661–3671. IEEE, 2021.

Wen Xia, Hong Jiang, Dan Feng, Fred Douglis, Philip Shilane, Yu Hua, Min Fu, Yucheng Zhang, and Yukun Zhou. A comprehensive study of the past, present, and future of data deduplication. *Proceedings of the IEEE*, 104(9):1681–1710, 2016.

Andrew Yale, Saloni Dash, Ritik Dutta, Isabelle Guyon, Adrien Pavao, and Kristin P Bennett. Generation and evaluation of privacy preserving synthetic health data. *Neurocomputing*, 416:244–255, 2020.

Shen Yan, Di Huang, and Mohammad Soleymani. Mitigating biases in multimodal personality assessment. In *International Conference on Multimodal Interaction (ICMI)*, page 361–369, 2020.

Show-Jane Yen and Yue-Shi Lee. Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications*, 36(3):5718–5727, 2009.

De Jong Yeong, Gustavo Velasco-Hernandez, John Barry, and Joseph Walsh. Sensor and sensor fusion technology in autonomous vehicles: A review. *Sensors*, 21(6):2140, 2021.

Dingwen Zhang, Junwei Han, Gong Cheng, and Ming-Hsuan Yang. Weakly supervised object localization and detection: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(9):5866–5885, 2022.

Jing Zhang, Xindong Wu, and Victor S. Sheng. Learning from crowdsourced labeled data: A survey. *Artificial Intelligence Review*, 46(4):543–576, 2016.

Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(11):1330–1334, 2000.

Zehui Zhao, Laith Alzubaidi, Jinglan Zhang, Ye Duan, and Yuantong Gu. A comparison review of transfer learning and self-supervised learning: Definitions, applications, advantages and limitations. *Expert Systems with Applications*, 242:122807, 2024.

Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National science review*, 5(1):44–53, 2018.