Artificial Intelligence Ethics and Safety

Practical tools for creating "good" models

Nicholas Kluge Corrêa[1]

AIRES PUCRS

[1] Master's degree in Electrical Engineering and doctoral student in Philosophy – PUCRS. Fellow of the Academic Excellence Program (Proex) of the CAPES Foundation (Coordination for the Improvement of Higher Education Personnel). President of the AIRES PUCRS Chapter.

Preface

> “Someday a computer will give a wrong answer to spare someone's feelings, and man will have invented artificial intelligence”.
>
> — Robert Breault

The **AI Robotics Ethics Society (AIRES)** is a non-profit organization founded in 2018 by Aaron Hui to promote awareness and the importance of ethical implementation and regulation of AI.

AIRES is now an organization with chapters at universities such as UCLA (Los Angeles), USC (University of Southern California), Caltech (California Institute of Technology), Stanford University, Cornell University, Brown University, and the Pontifical Catholic University of Rio Grande do Sul (Brazil).

AIRES at PUCRS is the first international chapter of AIRES, and as such, we are committed to promoting and enhancing the AIRES Mission. Our mission is to focus on educating the AI leaders of tomorrow in ethical principles to ensure that AI is created ethically and responsibly.

As there are still few proposals for how we should implement ethical principles and normative guidelines in the practice of AI system development, the goal of this work is to try to bridge this gap between discourse and praxis. Between abstract principles and technical implementation. In this work, we seek to introduce the reader to the topic of AI Ethics and Safety. At the same time, we present several tools to help developers of intelligent systems develop "good" models. This work is a developing guide published in English and Portuguese. Contributions and suggestions are welcome.

Introduction

"By far the greatest danger of Artificial Intelligence is that people conclude too early that they understand it".

— Eliezer Yudkowsky

It is not an uncommon situation when an individual, or a group of individuals, finds themselves in front of a decision-maker responsible for making some form of judgment based on a set of observable facts and characteristics (e.g., a judge in a civil court, an appraiser in a job interview, or a bank manager responsible for authorizing, or not, a loan). However, what is new is the use of statistical inference models to automate such processes (e.g., models created by machine learning).

As autonomous systems affect more and more people and society, understanding the potential risks related to such systems (and how to mitigate them) must be deepened. To anticipate, prevent, and mitigate the undesirable consequences of such systems, it is critical that we understand when and how problems can be introduced throughout the life cycle (i.e., data collection, training, validation, testing, deployment, etc.) of such systems.

We certainly cannot reduce all types of intelligent systems, or "Artificial Intelligence" (AI), to just Machine Learning. We also have the symbolic approach (Newell, 1990), the connectionist approach (Churchland & Sejnowski, 1992), hybrid methodologies (symbolic/connectionist), the mathematical-universal approach (Hutter, 2005), among several other methodologies that seek to develop systems capable of simulating certain cognitive capabilities to solve various types of problems (e.g., genetic algorithms, dynamic programming, BDI agents, etc.).

However, machine learning is currently one of the most widely adopted and used methodologies for various applications, especially deep learning with its different techniques (e.g., supervised, semi-supervised, unsupervised, self-supervised learning). We will focus in this guide mainly on the problems we face when developing applications that use this methodology.

Problems and side effects that arise from techniques such as reinforcement learning (Amodei et al., 2016), and risks related to potential advanced AI systems created by machine learning (Hubinger et al., 2019), will not be addressed in this guide.

As much as reinforcement learning has not yet "reached the mainstream," it is definitely a methodology capable of generating intelligent solutions, being the closest paradigm to what we may come to call "genuine AI" or "Artificial General Intelligence" (AGI).[2]

Certain advances are still needed to make reinforcement learning the new paradigm for machine learning solutions (e.g., efficient methods for developing reward functions or improving sampling efficiency). These advances are steadily and progressively being achieved (Ye et al., 2021). However, problems related to reinforcement learning (e.g., reward hacking, safe exploration, correctability) may soon manifest themselves in real-world applications.

However, such problems will not be the focus of this work. Here we will take a "short-term" view of problems involving AI ethics and safety, i.e., problems we face today, with the systems we own and use.

Systems created by machine learning (specifically supervised learning) learn statistical inference models based on observed datasets, in order to generalize their classifications/predictions/decisions to new data.

[2] In fact, for researchers such as Silver et al. (2021), the generic goal of maximizing reward may be enough to produce most of the intelligent behaviors studied in artificial and natural intelligence.

However, these systems can often create models that carry various types of biases or even act in undesirable ways:

- Facial recognition systems may exhibit racist biases (Lohr, 2018; Nunes, 2019);
- NLP (Natural Language Processing) systems can have sexist and misogynistic biases (Wolf et al., 2017; Balch, 2020);
- Classification systems may discriminate against members of the LGBTQ+ community (Wang & Kosinski, 2017; Agüera y Arcas et al., 2018).

Let us explore further the example that concerns racial discrimination: in October 2019, the then-current Minister of Justice Sergio Moro presented Ordinance No. 793 as a way to modernize Brazilian police forces. Nunes (2019) points out that since the implementation of such systems, the black population has been disproportionately affected. In 2019, 90.5% of individuals arrested caught by facial recognition and video-monitoring systems were black, the state of Bahia leading the number of arrests through these new technologies (51.7%), followed by Rio de Janeiro (31.7%), Santa Catarina (7.3%), Paraíba (3.3%) and Ceará (0.7%).

According to a report made available by the Criminal Defense Coordinator and the Access to Justice Studies and Research Directorate of the Rio de Janeiro Public Defender's Office,[3] between June 1, 2019, and March 10, 2020, there were at least 58 cases of mistaken image recognition, resulting in wrongful charges and even the imprisonment of innocent individuals. Of all those wrongfully accused, 70% were black.

[3] Public Defender's Office of the State of Rio de Janeiro. http://www.defensoria.rj.def.br/uploads/imagens/d12a8206c9044a3e92716341a99b2f6f.pdf.

But why is this so?

A simplistic answer would be, “The answer is in the data. The data we use is skewed.” However, a more truthful answer would be, “It's a complex problem.”

There is much that we still do not understand about such systems. At the 2017 NIPS conference (Conference on Neural Information Processing Systems), Ali Rahimi[4] raised an important point about the current state of the Machine Learning research field: “machine learning has become alchemy.” “In the old days” (i.e., before deep learning became the paradigm), techniques such as linear regression, logistic regression, support vector machine, guaranteed efficient and interpretable solutions. However, we were not able to use such methodologies for more complex problems (e.g., computer vision).

However, it is important to remember that, as much as machine learning generates statistical inference models, machine learning *is not like statistics.* In machine learning, we have many hypotheses and few theorems. This is why machine learning is closer to a discipline like engineering than mathematics. We figure out what works by trial and error. We don't yet have theorems that allow us to rigorously verify the behavior of such systems and thus make predictions about how they might behave in the future (OOD - “out-of-distribution”).

In Ali Rahimi's words:

> Alchemy is not bad. There is a place for alchemy. Alchemy “worked.” Alchemists invented metallurgy, ways to dye textiles, our modern glass making processes and medications. Then again alchemists also believed that could cure diseases with leeches and transmute base metals into gold. For the physics and chemistry of the 1700s to usher in the sea change in our understanding of the universe that we now experience, scientists had to dismantle 2,000 years’ worth of alchemical theories. If

[4] Transcript available at: https://www.zachpfeffer.com/single-post/2018/12/04/transcript-of-ali-rahimi-nips-2017-test-of-time-award-presentation-speech.

> you're building photo-sharing systems alchemy is okay. But we're beyond that now. We're building systems that govern healthcare and mediate our civic dialogue. We influence elections. I would like to live in a society whose systems are built on top of verifiable, rigorous, thorough knowledge and not on alchemy.

In other words, machine learning still needs more theoretical study. However, it is not clear that the industry will slow down its practical progress and development for the sake of caution and formalization of the theories that underlie the creation of its products. And that creates problems.

Thus, we believe that it is necessary to create and formalize a new agent to operate within organizations and companies focused on developing technologies and solutions that use these types of systems. We need security engineers and ethicists who specialize in machine learning, i.e., agents responsible for preventing and mitigating the possible side effects of systems created by machine learning.)

Such an actor would be responsible for helping to implement security measures during the entire life cycle of such systems to ensure that certain ethical principles are respected and implemented during the development, deployment, and monitoring of such systems.

To meet this need, one of the responses proposed by the community involved in the field of Artificial Intelligence Ethics, such as government institutions, private corporations, academic institutions, civil societies, professional associations, and NGOs, has been the publication of several principled governance mechanisms. These mechanisms can be defined as codes of ethics, guidelines, among other similar governance instruments, that is, normative documents based on ethical principles (Russell et al., 2015; Boddington, 2017; Goldsmith & Burton, 2017; Floridi et al., 2018; Greene et al., 2019).

AI Ethics, as much as it is a relatively new field of Ethics,[5] has enough literature that meta-analyses of the field have been conducted (Jobin et al., 2019; Hagendorff, 2020; Fjeld et al., 2020). These meta-analyses point out that there is a convergence towards a certain group of commonly held ethical principles (values):

Values	**Description**
Transparency	This principle points out one of the biggest deficits in contemporary Machine Learning techniques. While humans expect explanations they can understand, machine learning algorithms operate on complex statistical computations that defy simple translations, making them "opaque" (Mittelstadt et al., 2019).
Justice/Equity	Issues of fairness include problems of equal treatment and fair distribution of benefits. This principle is generally worked out in the literature through algorithmic definitions of fairness and equity (e.g., Statistical/Demographic Parity, Predictive Parity, Equalized Probabilities) (Galhotra et al., 2017; Verma & Rubin, 2018).
Privacy	Data is like coal for the AI industry. And the big tech companies (Google, Amazon, Facebook), are the new "coal mines" of the 21st century. The abundance of data we produce daily guarantees an almost inexhaustible source of information for training AI systems. However, the use of personal data without consent is one of the main concerns found in the literature (Ekstrand et al., 2018).
Accountability	How to make the AI industry accountable for its technologies. For example, in the case of autonomous vehicles, what kind of guarantees and responsibilities should companies developing autonomous vehicles provide to their customers and society at large (Maxmen, 2018)?
Reliability	Reliability is an ethical principle close to transparency. This principle defends the idea that AI systems should be robust. Depending on the type of model, and context that such a model is embedded in (e.g., automating judicial system

[5] According to Jobin et al. (2019), less than 20% of all AI Ethics documents reviewed in their meta-analysis (84) are more than four years old. According to the NGO AlgorithmWatch (2020), their Global Inventory of AI Ethics Guidelines contains 173 documents. None of these documents predate the year 2013. Of these documents, only two have their origin tied to South Africa and South Asia (no documents produced by Latin America are listed).

	decision-making), it is of paramount importance that such systems are resilient to, for example, adversarial attacks (Krafft et al., 2020).
Beneficence/Non-Maleficence	This principle advocates that artificial intelligence be used to promote "Good." Since "Good" is a difficult concept to specify, many consider non-maleficence (e.g., AI should not cause harm) as a better specification. This principle is very close to what we call AI Safety (Amodei et al., 2016).
Freedom/Autonomy	This principle advocates the idea that freedom/autonomy (i.e., the experience that we own and are responsible for our own choices and preferences) is fundamental to human psychological well-being. AI systems should not remove our autonomy, but rather empower it (Calvo et al., 2020).
Dignity	This principle refers to the inherent value (and inherent vulnerability) of the human individual. Something that should be (human dignity) inviolable. AI systems should be developed to promote an ecosystem that ensures that individuals are seen, heard, listened to, treated fairly, recognized, understood, and feel safe (Ruster, 2021).
Sustainability	This principle can be understood as a form of "intergenerational justice." Sustainability describes our ethical obligation to future generations. An obligation to secure and preserve their living conditions, by, for example, through the careful use of our natural resources (Krafft et al., 2020).
*Solidarity**	This principle can be understood as sharing the prosperity created by AI. We must implement mechanisms to redistribute the increased productivity, share new burdens and responsibilities, and make sure that AI will not increase the inequality of our world (Luengo-Oroz, 2019).
*Diversity**	This principle can be understood as the defense and valorization of the different ways in which the human entity can come to express itself, by any group or identity it wishes. AI systems should be developed in a way that protects and values our diversity (AIRES at PUCRS, 2021).

*Inclusion**	AI systems should be developed in such a way as to "include," not exclude. This principle advocates for the welcoming of all forms in which the human entity can come to express itself, regardless of specific affiliations, groups, or identities (AIRES at PUCRS, 2021).

* Regarding the principles of Solidarity, Diversity, and Inclusion: these are the principles least raised by the current state of AI Ethics. However, we feel it necessary to point them out as important and include them within this short and incomplete list.

At the same time, in 2017, the IEEE Standards Association published the second version of the document *"Ethically Aligned Design: A Vision for Prioritizing Human Well-being With Artificial Intelligence and Autonomous Systems."* Such a document suggests several methodologies to guide ethical research in projects seeking artificial intelligence development, upholding the human values outlined by the United Nations Universal Declaration of Human Rights. The document even guides certain guidelines and recommendations for the development of *"ethically aligned AI"* (73-82). [6]

However, there are several criticisms raised against this type of abstract principle-based methodology (Principialism), which without a translation into the practice of intelligent system development, risks being categorized as mere "ethics theater," i.e., a moral discourse with little (or no) intention of solving real-world problems (Calo, 2017; Rességuier & Rodrigues; 2020; Corrêa & De Oliveira; 2021).

In the words of Mittelstadt (2019, p. 503):

> Statements based on vague normative concepts hide points of political and ethical conflict. "Justice," "dignity," and other abstract concepts are examples of "essentially contested concepts" that have many possible conflicting meanings that require contextual interpretation [...] At best, this conceptual ambiguity allows for a context-sensitive specification of ethical requirements for AI. At worst, it masks fundamental principled disagreements and

[6] The IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems. Ethically Aligned Design: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems, Version 2. IEEE, 2017. http://standards. ieee.org/develop/indconn/ec/autonomous_ systems.html.

> leads AI ethics toward moral relativism. At the very least, any compromise reached so far around fundamental principles for AI ethics does not reflect a meaningful consensus on a common practical direction for the "good" development and governance of AI.

Thus, it is important to note that there are still few proposals for how we should implement ethical principles and normative guidelines in the practice of AI system development. The goal of this handbook is to try to bridge this gap between discourse and praxis. Between abstract principles and technical implementation.

To begin this guide, in the next section, we will investigate what the development cycle of machine learning trained models looks like to better understand where problems might arise.

The “Life Cycle” of a Machine Learning System

The “life cycle” of a model trained by machine learning, i.e., the design, creation, deployment, and monitoring of such a system, cannot be isolated from human interference, as such systems are designed, deployed, used, and monitored by humans. Ananny & Crawford (2018) suggest that algorithmic systems are sets of human and non-human actors entangled in a dynamic that generates non-deterministic effects, and in our opinion, this is an excellent definition.

Thus, to understand and prospect the ethical implications of such systems, it is necessary to understand how the whole system functions, i.e., the total set of human and non-human actors that compose it.

Systems created by machine learning (especially supervised learning and its variants) generally follow the following cycle:

- *Data collection and pre-processing:* before any analysis or learning can take place, we need data. The dataset for training is usually created from two assumptions: (1) you assume that your outputs can be predicted given your inputs; (2) you assume that the available data is informative enough to learn the relationship between inputs and outputs. Sometimes we can accept such assumptions (e.g., shoe size is generally related to an individual's height), and sometimes we cannot (e.g., the historical value of some asset, such as Apple stock, may not correlate with its future value). If you want to create a system to predict the chance that a patient will develop diabetes, you could create a database based on the users of the public health network that have type 2 diabetes. You could use certain characteristics (features) that you believe are

correlated with type 2 diabetes (e.g., weight, age, BMI, family history, physical activity) together with samples of people who have type 2 diabetes (labeled data) to create a model that, given the input values you have established, predicts the chance that this individual will or will not develop type 2 diabetes. Almost 80% of all the work of a machine learning engineer is in creating a good dataset;

- *Model Development:* after defining our dataset, we divide it into three groups: training, validation, and testing. It is considered "good conduct" (if not common sense) not to mix the training dataset with the dataset that will be used for validation and testing. What we want is a model that generalizes its predictions to new samples, not a model that simply memorizes the presented data (i.e., overfitting). After this split, we define our model architecture (e.g., feedforward neural network), our objective function (e.g., predicting the individuals most susceptible to type 2 diabetes), our loss function (e.g., binary cross-entropy), optimizer (e.g., stochastic gradient descent), and evaluation metrics (e.g., accuracy, precision, recall, AUC). Next, we will iteratively adjust the parameters of our model (e.g., number of nodes in the hidden layer, learning rate of the optimizer, a different loss function) to improve the result of our evaluation metric on the validation data. In this step, we (re)train our model until we are satisfied with its result in the validation phase;
- *Model Evaluation:* we train the model with the training portion of the data and evaluate its performance for parameter fitting with the validation portion. Then, when we are satisfied with the final model, we evaluate it with the testing portion of the data. This is when we evaluate the predictive power of our model with data the

model has never seen before. We can also evaluate our model against benchmarks if these exist;

- *Post-processing:* at this stage, we need to adapt the output of our model to the problem we are dealing with. For example, if we define that the output of our model (i.e., the model that infers the chance of an individual developing type 2 diabetes) is a probability measure between 0 and 1, but we want a categorical answer (i.e., "Yes," "No," "Inconclusive"), we need to estimate a decision threshold (e.g., more than 80% confidence equals a conclusive classification);
- *Model Deployment:* in this stage, many adjustments should be made to make the model "ergonomic", facilitating its interaction with its users. For example, transparency tools can be implemented in the model in cases where interpretability is vital (e.g., the VICTOR system used by the Supreme Court). Deployment is always a sensitive moment in the life cycle of these systems. Since the training environment is (usually) not a faithful representation of the deployment environment, unexpected/unwanted behavior may occur just at this stage (e.g., a clothing recommendation system that was naively trained in the summer and is unable to recommend "useful" garments during the winter);
- *Monitoring:* After the model is deployed, it is necessary to monitor its behavior to ensure that the system performs the function for which it was developed, and does not result in any kind of behavior that we might consider undesirable/unsafe (e.g., a disease prediction model that has a low performance for a specific group should, in theory, be taken out of circulation and improved).

With the development, deployment, and monitoring cycle of such a system defined, we will investigate in the next section "where" problems can arise and compromise a model created by machine learning.

Sources of Problems

As we can see from the previous section, many assumptions and choices will be made before we deploy our model to act in the real world. Developers, machine learning engineers, data scientists, all these actors will be actively influencing the form the model will take during its development, be it in building the datasets (training, validation, and testing), choosing and crafting the features (feature engineering), defining the model parameters, choosing the evaluation method, etc. During this long process, many "bad" decisions can negatively influence the final model.

As mentioned in the opening section, the source of these problems is not simply "skewed data." First of all, databases are not static structures, divorced from the social/historical contexts and intentions from which they arose. To mask the side effects generated by such systems as just "skewed data" is to obfuscate the complexity of how such systems can be compromised throughout their life cycle. At the same time, it is to obfuscate our share of responsibility for the problem.

Suresh & Guttag (2021), in their study *"Understanding Potential Sources of Harm throughout the Machine Learning Life Cycle,"* provide a framework that identifies seven distinct sources of harm that can compromise the behavior of such systems, from data collection to deployment:

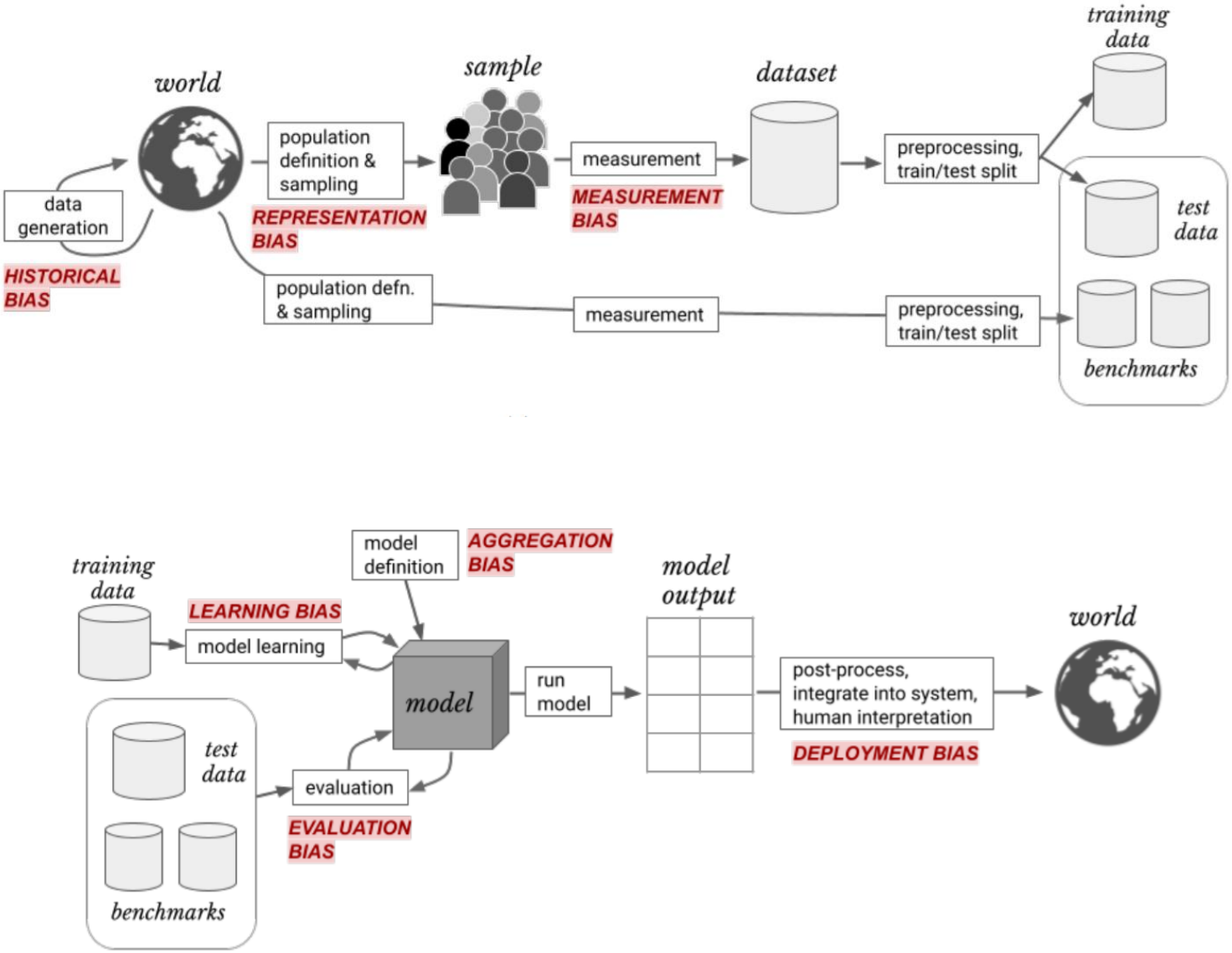
- *Historical biases:* this kind of problem occurs because our world, as it is or was, is flawed. Thus, even if the model is a perfect representation of the environment, it may still generate harm because it represents an imperfect environment. For example, Brown et al. (2020, p. 36-37) report that their model of language (GPT-3) associates pejorative, sexist, and misogynistic adjectives

more often with women than men (i.e., a reflection of the texts, and culture, that we encounter on the Internet);

- *Representation biases:* This problem occurs when the data used to train the model does not represent the population or environment in which the model will operate. When training a model to predict the development of type 2 diabetes, there may not be enough data to represent all possible groups of interest (men, women, elderly, children, etc.). Or, image recognition software used by an autonomous car trained in urban environments may operate flawed when operating in rural regions;
- *Measurement biases:* when we choose characteristics (features) to be used for some outcome (e.g., BMI, weight, height, family history), we are assuming that such a characteristic/feature is representative of what we want to predict or classify (type 2 diabetes). But this is not always the case, as such "proxies" may only be approximations of a more complex reality. For example, if our model gives too much weight to the variable "BMI" for the task of predicting type 2 diabetes, subjects with a large amount of muscle (e.g., bodybuilders) might be falsely classified as potential developers of type 2 diabetes. "IQ" (i.e., intelligence quotient) may not be a good parameter for assessing academic success, which often depends on other factors that are difficult to measure (e.g., motivation, ability to relate, organizational skills);
- *Aggregation biases:* This type of problem occurs when different groups are joined into a single dataset. However, the trained model does not perform efficiently with any (or none) of the groups. For example, it is known that men are twice as likely as women to have a heart attack in their lifetime. A model trained on a mixed dataset (i.e. without differentiating the sex of the samples) to predict the chance of a patient having a heart attack, may turn out to be inefficient with either (or both) sexes. Ideally, models should be trained to fit specific groups (when necessary);

- *Learning biases:* the choice of the loss function (e.g., root mean square error, binary cross-entropy, categorical cross-entropy) and performance metric (e.g., accuracy, precision, recall, AUC) can influence the type of output our model generates, and how we interpret its performance. For example, if we have an application for which false-negative classifications can generate a "large cost" (e.g., false negative for HIV), perhaps we should not use accuracy to measure its performance, but rather recall;
- *Evaluation biases:* the data used in the testing (or test-benchmark) phase does not always represent a good evaluation metric for the domain in which the model will be deployed. For example, you may have developed a face recognition model with excellent performance in your testing phase. However, the benchmark you used has a low representation of the brown population (e.g., 4%), and the model will perform in a domain where much of the population is brown (e.g., Brazil);
- *Deployment biases:* this kind of problem occurs when the model is used differently (or beyond) what it was originally developed to do. Many of the models created by machine learning are not "fully autonomous," but are found as part of a socio-technical process where human intentions and desires are part of it. For example, risk assessment systems are used in the American penal system to predict the likelihood that a person will commit a future crime (i.e., criminal recidivism). However, a perverse instantiation of this tool would be to use it to determine the length of a sentence based on the likely risk of recidivism (Collins, 2018).

Below is a diagram describing the life cycle of a model developed by machine learning and where the biases described above manifest themselves.



Sources of problems during the development and deployment of a model (Suresh & Guttag, 2021).

It is important to note that depending on the application of a model, the types of problems mentioned above will not manifest themselves in any detrimental way, making an “ethical analysis” unnecessary. For example, a model created to optimize industrial processes (e.g., quality control, inventory control), which do not affect people's lives in a significant way, does not require the same level of analysis that models that interact

directly with people do. In other words, there is, for example, no "breach of privacy" in situations where a model must classify a fruit as "fit for consumption" or "not fit for consumption."

However, if during an initial inspection it is revealed that there are ethical issues to consider, the organization and developers responsible should conduct a full ethical evaluation of the model.

In short, *context is what will define the standard*. There is no "One True Moral Theory" that applies to every application of an algorithmic model.

In the next section, now that we know several types of problems that can interfere with the development and deployment of a model created by machine learning, we will explore some of the definitions of algorithmic "fairness" (sometimes also referred to as "equality" or "fairness").

Defining “Fairness” in Machine Learning

Taking all the examples of biases cited in the last section, what we would ideally like to do is develop a “fair” model, i.e., a model that performs its function free of discrimination and bias. There is a growing body of work on “fair algorithms” being published, and we can define “fair algorithms,” in the context of machine learning, as a model that satisfies some particular notion of “fairness.”

However, depending on how we formalize “fairness” or “justice,” different decisions/classifications/predictions will be defined as “fair.” These decisions may conflict with other particular formalizations of “what we mean by fair”:

- Fairness means achieving parity among the demographic groups in a population;
- Fairness means satisfying the preferences of the demographic groups in a population;
- Fairness means equally benefiting all demographic groups in a population;
- Fairness means impacting (i.e., opposite of benefiting) equally all demographic groups in a population;
- Fairness means judging from behind the veil of ignorance;

What would be the best definition to apply in the context of machine learning? First, we need to define fairness, in its various forms, in statistical terms, i.e., how the statistical inferences of a model can be performed in a way that respects specific notions of fairness. Let's look at some of these possible definitions:

- *Veil of Ignorance:* a model satisfies this condition if all the sensitive attributes (i.e., attributes for which non-discrimination should be established) of its samples are not made explicit to the model, i.e.,

the model has no access to information such as race, ethnicity, color, national origin, sex, sexual orientation, etc.

This approach can be traced back to the definition of Justice advocated by John Rawls (1999) in his seminal work "A Theory of Justice." One of the problems with this approach is that we need to define which proxies can be used by a model to identify (and discriminate) samples. Even if sensitive attributes are veiled from the model, a predictor could still infer and discriminate marginalized populations based on non-sensitive information. For example, if a bank uses a model to assist in evaluating credit card applications, and the city/region where this back provides service has a certain level of racial segregation in its distribution of residents, non-sensitive attributes (e.g., zip code) could be used to discriminate against individuals living in certain locations.

Moreover, certain studies point out that the veil of ignorance may turn out to be more discriminatory than "fairness by awareness" (i.e., when we take sensitive attributes into account in a judgment) (Sen, 1990; Bonilla-Silva, 2003; Fryer et al., 2008). At the same time, this approach seems to go against principles of "repair and relief" of historically marginalized populations.

- *Fairness by Awareness:* a model satisfies this condition if the model produces the same output for similar individuals. That is, if two samples have a minimum number of similar features, both samples will be classified equally.

This definition is a more elaborate variation of the previous criterion ("Veil of Ignorance"), where we define a model as "fair" that treats similar individuals similarly. To put this definition into practice, we first need to: (1) define a distance metric for us to measure the similarity between two samples; and (2) define what is the minimum distance for two samples to be similarly classified. For example, a possible distance metric f could

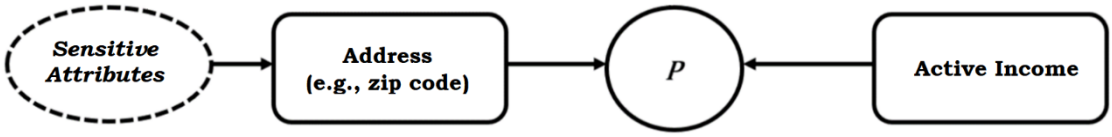
define that the distance between two subjects, i and j, as 0 if all non-sensitive attributes are identical and 1 if any non-sensitive attributes are different.

However, it is not at all clear how to create a function that takes as an argument the features of two samples and calculates their distance without generating discriminative classifications. By this definition, we just "pass the problem" of making a fair classification from the model to the distance metric.

- *Counterfactual equality:* a model satisfies the counterfactual equality condition if its ranking is unchanged, even if the sensitive attributes of the sample were different.

This is the same thing as saying that the model is "fair" if, for example, it authorizes the opening for a line of credit for subject A, who possesses the sensitive attribute i, even if subject A possessed the sensitive attribute j.

To implement a model that satisfies this condition, we first need to determine what a "counterfactual influence" would be (i.e., how sensitive attributes can influence non-sensitive attributes to determine the outcome of a classifier). One way we can do this type of analysis is by using causal graphs/diagrams (Pearl, 1995; Kilbertus et al., 2017).



In the graph above, sensitive attributes (e.g., race) could be inferred by "Address," which would act as a proxy for sensitive attributes in a city where racial segregation occurs in the distribution of its residents. Thus, we can define that a causal model is fair by counterfactual equality if no sensitive attributes can influence a proxy that directly affects the final decision (P) of the model (the model above does not satisfy this condition).

To satisfy this fairness condition, it would be necessary not to use (as classification features) all the descendants of nodes containing sensitive attributes. However, in real applications, the vast majority of sensitive attributes are nodes whose descendants span through almost the entire causal graph.

- *Statistical/demographic parity:* a model satisfies the statistical/demographic parity condition if the model can generate equal, or nearly equal, results for members of groups with different sensitive attributes.

Statistical parity for groups can be referred to as a notion of egalitarian-collectivist distributive justice (Hofstede et al., 2010; Hirose, 2014). One criticism raised against this formalization is that by imposing statistical parity on our model, we may generate: (1) a model that produces incorrect results (i.e., the model will prioritize statistical parity over its performance); or (2) the model may come to discriminate against more "qualified" individuals (Luong et al., 2011, Dwork et al., 2012). For example, a model designed to aid in the selection process for a job opening might prioritize statistical parity over other relevant attributes (e.g., "gender over resume").

Thus, we must understand that imposing fairness metrics (e.g., statistical/demographic parity) should be seen as a compromise between "performance" and "fairness," since we will be intentionally introducing impartialities that deviate from the original data distribution.

For certain applications, this would not be desirable. For example, a sentiment classifier (a common machine learning task in NLP) is biased to classify words such as "suicide," "depression," "loneliness," as text containing negative sentiment, and that is desirable. However, models biased in favor, or detriment, of social classes, races, or genders, are not desirable.

We can also define statistical parity in terms of predicted outcome:

- *Predictive Parity:* A model satisfies the predictive parity condition if the model's precision is equal across different groups. That is, if the model determines with 90% accuracy that an individual is a "good candidate for a loan" (i.e., the loan will not be harmful to the bank or the subject), this measure of accuracy must be independent of the value of sensitive attributes.

One of the difficulties in implementing a model that satisfies the condition of predictive parity between different groups is that groups are not always equally represented in datasets. For example, of the large public face image datasets (e.g., UTKFace, CelebA, LFWA+), there is a strong bias in favor of Caucasian faces, while other races (e.g., indigenous) are significantly underrepresented (Kärkkäinen & Joo, 2019). For a model to achieve balanced performance across groups, there must be enough examples for the system to learn a good model.

- *Equalized Odds:* this algorithmic fairness condition can be interpreted as an extension of the predictive parity condition. A model that satisfies this condition is a model that has an equally true and false positive rate, regardless of the value of sensitive attributes.

This means that the chance of an individual with a good credit score receiving a positive rating (i.e., being approved for a new credit card), and the chance of an individual with a bad credit score receiving a positive rating, is equal and independent of the group (i.e., sensitive attributes) to which that individual belongs. In other words, both members of *i* and *j* have the same chance of receiving a positive rating (whether it is correct or not).

Certainly, these are not the only existing definitions of algorithmic fairness, and other definitions can be found in the literature on "machine learning fairness" (Chouldechova, 2016; Hardt et al., 2016, Corbett-Davies et al., 2017; Galhotra et al., 2017; Kilbertus et al., 2017; Verma &

Rubin, 2018, Gajane & Pechenizkiy, 2018, Mehrabi et al., 2019). However, the point we wish to make clear is:

- *There is no single definition of what is fair.*

Organizations concerned with developing "fair" AI systems (i.e., capable of mitigating the emergence of certain biases throughout the life cycle of the model created) must first establish which definition of "fair" best applies to the problem their model will tackle. Certain definitions may be more applicable to certain contexts. For some applications, it may be important to "obscure" all forms of sensitive attributes, while for others, it may be better to prioritize statistical parity over classification efficiency.

In short, the correct definition of "fairness" depends on the context. However, wouldn't it be possible for us to apply all the suggested definitions as fairness constraints to the same model? Unfortunately, there are limitations on how we can constrain the predictions of statistical inference models. Let us look at some of these constraints in the next section.

Impossibility Results in AI Ethics

Many of the definitions presented in the last section may seem similar or variants of one of the same general goal, i.e., that the inferences of a probabilistic model be independent of certain sensitive attributes, such as gender, race, sexual orientation, etc. However, when we try to calibrate our model so that it satisfies multiple notions of statistical equality and parity, we arrive at certain impossibility results.

Since 2016, thanks to the work of Kleinberg et al. (2016), we already know that certain notions of justice, in the context of probabilistic classifications, are incompatible with each other, i.e., we cannot satisfy all of them at the same time. There are certain inevitable arbitrages between different definitions, regardless of the specific context and method used to arrive at a probabilistic classification.

In machine learning, when we design a model for classification/prediction purposes, we use a group (x_i) of labeled data (y_i) to train our model. The goal that our model must fulfill is to find a function $f: X \to Y$ that approximates the true joint distribution of samples and labels ($X \times Y$). Thus, the goal of the model can be defined in terms of minimizing empirical risk, i.e., decreasing the gap between the model's predictions and the true labels of its samples.

Statistical modeling for empirical risk minimization can be thought of as a condition orthogonal to any notion of fairness that causes the model to deviate from the true joint distribution of samples and labels. In other words, standard loss minimization (e.g., binary cross-entropy) and optimization (e.g., stochastic gradient descent) techniques seek to minimize empirical risk, not to adhere to particular notions of fairness. Thus, a fair classifier would narrow the gap between “fair predictions” and “empirical predictions” (Saravanakumar, 2021).

When designing a fair classifier, the problem we want to avoid is that sensitive attributes interfere with the classification of our model. We can say that a sensitive attribute (a) is a possible source of bias, only if such an attribute is statistically correlated with the prediction ($\hat{y}$) of our model ($P_a[\hat{y}] = P(a)$). Otherwise, the sensitive attribute will not interfere with the inference of the model in question because it is not correlated with the prediction, or true value, of the sample being considered.

From what we can see in the last section, defining a "fair" statistical inference algorithm is equivalent to defining some calibration criterion that fits any of the various definitions of "fairness," "equity," and "equality" found in the literature. The impossibility results of Kleinberg et al. (2016) apply to three of these definitions, dictating that barring ideal cases, no statistical inference model can satisfy the following three calibration criteria:

- **A.** *Calibration within groups:* a model satisfies this condition if for each possible group (e.g., i and j), the model classifies the members of i and j that satisfy the positive condition into a given class with the same chance, i.e., both individuals from group i and j that have the same probability of being positively classified, will have the same chance of being positively classified by the model.
- **B.** *Balance for the positive class:* a model satisfies this condition if the chance of an individual being classified to the positive class is independent of his or her group. A model that does not satisfy this condition is a model that privileges (i.e., positively classifies with a greater chance) members of one group (i) over the other (j).
- **C.** *Balance for the negative class:* this is the inverse condition of the previous definition. A model satisfies this condition if the chance of an individual being classified as a negative class is independent of

his group. Corollary, a model that does not satisfy this condition is a model that privileges members of one group over the other.

Criterion A can be defined as a more restricted form of the statistical/demographic parity condition. Criteria B and C can be interpreted as versions of the Predictive Parity and Equalized Odds conditions. These three definitions of algorithmic justice are some of the most accepted and studied by the AI Ethics community, and these are also the victims of this impossibility result.

According to the impossibility theorems of Kleinberg et al. (2016), there are only two exceptions to this rule:

- *Ideal Cases:* the only examples of problems in which there is a probabilistic classification that satisfies fairness conditions A, B, and C, are when: (1) the inference model is perfect (i.e., the joint distribution of $X \times Y$ is perfectly known, and $P[\hat{y}]$ is equal to 0 or 1 for all x_i); or (2) the inference model has equal base rates (i.e., the chance of a sample being classified as belonging to positive and negative class is equal).[7]

Unfortunately, both ideal cases are not the "norm" in terms of probabilistic models created by machine learning. If we knew the perfect distribution of all possible samples, we would not need machine learning because we have access to an oracle. And a model with an equal balance ratio between negative and positive classes will generally not represent the true data distribution. Various situations and applications cannot be decided by the toss of a fair coin (even if that is, "statistically," the fairest thing to do).

Let's use again our example of an individual who goes to a bank to try to open a new line of credit, and one of his evaluations will be performed by

[7] Kleinberg et al. (2016) also proved that their impossibility results can be extended to situations where we only approximate the ideal cases, i.e., the model only approximates a perfect prediction with an error $\varepsilon > 0$, or the model only approximates an equal ratio between the total balance between negative and positive classes, for any $\delta > 0$.

a statistical inference model (e.g., a supervised machine learning model), which will result in his "credit score".

The sampling distribution is probably not uniform, i.e., the environment is not made up of 50% individuals with a good credit score (i.e., a new line of credit would be beneficial to both the bank and the individual) and 50% individuals with a bad credit score (i.e., a new line of credit would be harmful to both the bank and the individual). Similarly, the distribution of sensitive attributes between the two classes (for simplicity, we are imagining a binary classification problem) is probably not uniform (i.e., perhaps the actual distribution of "individuals with a good credit score" favors women).

Thus, what the impossibility results tell us is that except the two types of ideal cases, at least two of the following undesirable properties must hold, because no probabilistic inference model can simultaneously satisfy calibration criteria A, B and C, i.e., we are only able to satisfy one criterion at the expense of two:

1) *Statistical/demographic parity violation:* the results of the classifier/predictor/model are systematically biased upwards or downwards for at least one group;
2) *Predictive Parity Violation:* the rate of ratings for the positive class is systematically biased, assigning higher probability to the positive class for at least one group;
3) *Equalized Probability Violation:* the rate of ratings for the negative class is systematically biased, assigning a higher probability for the negative class for at least one group.

The tradeoff between these three conditions is not necessarily a machine learning problem, but a fact about probabilistic classification problems that seek to model data produced by real-world phenomena. This impossibility should not be attributed to a lack of model capability, but

rather to the constraints of the data generation regime and the conditions of equality and fairness that we stipulate. Another way to interpret this result is that the algorithmic justice problem is not exactly a statistical problem, but a sociological problem, since the discrepancies and biases embedded in the data are merely reflections of an unequal and imperfect society.

Thus, a machine learning engineer who builds models for applications with possible social impacts should be prepared to deal with this phenomenon. *To choose a metric of fairness is also to choose which violations we are willing to do.*

There is no general solution to this problem. It is the responsibility of the developers and supervisors of a project that aims to create such models to define by which ruler to norm their system. However, what should be done is: (1) investigating the limitations and possible biases of the model generated; (2) making such information available (transparently) to those who will use (be impacted by) such technologies.

A bank manager assisted by an AI model should know if this is the case, that his model has certain biases in its decision making. Such biases, and the measures and choices that have been made to mitigate their possible side effects, should be explicitly made available to the operator. For example, perhaps the model could come with a “package insert” or “letter” explaining the possible biases that the model may exhibit. When a rating for sample X is made, perhaps the model could result in not only a rating but also a warning (“Warning! This model tends to generate a systematically biased False Negative percentage for samples with the following sensitive attribute: 'Divorced'.”).

In the next section we will start to present some possible solutions (tools and methodologies) to mitigate the problems presented so far.

The AI Robotics Ethics Society®

The Role of the AI Safety Engineer

Imagine you are in charge of the AI ethics and safety division of a company that produces solutions and products through machine learning techniques. Your duty is to (1) ensure that the models generated by your company follow certain safety protocols; (2) ensure that possible side effects are detected and predicted before the model is deployed to act in the environment; and (3) monitor the behavior of the model "in the wild." The problems cited in the last section are some of the concerns that should be on your radar:

- How are the various forms of oppression and historical biases characteristic of the context where the model will be deployed (i.e., historical biases) structured in our social and political fabric?
- Is the data used for training an accurate representation of the population or domain of interest? Are there important but marginalized groups that are not present in this dataset (i.e., representation bias)?
- Are the chosen labels and characteristics good proxies for what we are interested in measuring/classifying/predicting (i.e., measurement bias)?
- Given the problem we face, would it be correct to aggregate different groups? Or do we need to treat each group concerning its specificities (i.e., aggregation biases)?
- How can the model be used for purposes other than those defined by the developers (i.e., deployment biases)? What types of adversarial attacks is the model most susceptible to?
- What fairness metrics are being followed? What algorithmic fairness conditions does the model violate (i.e.,

Statistical/demographic parity, Predictive parity, Equalized probabilities)? Are algorithmic fairness constraints something necessary for the application in question?

Understanding where intervention is needed and how feasible it is can inform discussions about how damage can be mitigated versus when it is better not to deploy a system at all. Let's start exploring some qualitative tools to help developers perform such an analysis.

The AI Robotics Ethics Society®

Translational Tools

Translational tools, in the context of AI Ethics, are methodologies to help developers "translate" abstract, high-level, ethical principles into practical, concrete implementations. Floridi and Taddeo (2016) suggest that this type of tool can be thought of as a diagnostic methodology, i.e., a way to assess whether a given model is aligned with certain ethical principles espoused and defined by developers.

We will define these types of tools as "qualitative", and in the next sections we will present the following diagnostic tools:

- *FAIR (Findability, Accessibility, Interoperability, and Reusability);*
- *Digital Catapult AI Ethics Framework;*
- *VCIO (Values, Criteria, Indicators, Observables).*

Building a Fair Dataset (FAIR)

When the problems in our model can be traced back to the dataset we used, a possible solution is to correct such a dataset so that its sampling distribution, concerning sensitive attributes, respects some particular condition of fairness that we wish to implement.

For example, FairFace[8] is a dataset of faces with a balanced distribution across genders, races, and ages, containing 108,501 images. Kärkkäinen and Joo (2021) demonstrated that models trained with FairFace are significantly more accurate than other models trained with sets such as UTKFace, CelebA, LFWA+, showing consistent performance across groups (i.e., predictive parity across race, gender, and age).

To help developers identify and choose fair datasets, we can use the methodology proposed by Wilkinson et al. (2016): FAIR. The FAIR methodology is a tool for developers to evaluate certain characteristics of the dataset they intend to use, these being: Findability, Accessibility, Interoperability, and Reusability.

These principles serve to guide developers to ascertain three types of entities: (1) data (digital sources of information); (2) metadata (information about digital information); and (3) infrastructure (how the data and metadata are structured and indexed).[9] Let's look at some of the recommendations made by the FAIR methodology.

Findability, i.e., data and metadata must be easily accessible to both humans and computers:

- The (meta)data is assigned a globally unique and persistent identifier (e.g., a repository on GitHub, the “Orcid” of the

[8] https://Github.com/joojs/fairface.

[9] For more information on how to implement the FAIR methodology, please visit https://www.go-fair.org/.

responsible researcher, the "Doi" of a publication demonstrating the results of applications of the dataset);

- The data are described with rich metadata (e.g., DICOM: Digital Imaging and Communications in Medicine is a protocol for processing, storing, and transmitting medical information in electronic form to allow, for example, medical imaging information to be accessible between different diagnostic equipment, imagers, computers, and hospitals);
- Metadata clearly and explicitly include the identifier of the data they describe (e.g., the association between a metadata file and the dataset must be explicitly referenced in the metadata by a globally unique and persistent identifier);
- The (meta)data is recorded or indexed in a searchable resource (e.g., the dataset can be found by a public search engine, e.g., Google).

Accessibility, i.e., after the (meta)data is found, developers should know what procedures to use to gain access (e.g., authentication and authorization) to the dataset:

- The (meta)data can be retrieved by its identifier using a standardized communication protocol (e.g., the dataset can be accessed and downloaded by an HTTP link);
- The protocol is open, free, and universally implementable (e.g., the dataset is free);
- The protocol allows an authentication and authorization procedure when needed (e.g., datasets must have their access conditions explicitly stated, e.g., authentication by phone number);
- Metadata is accessible even when the data is no longer available (e.g., if the dataset can no longer be accessed by its identifier,

metadata referring to that dataset will make it explicit that the dataset has “expired its useful life”).

Interoperability, i.e., the data set must be in a format that allows its integration with various platforms and applications:

- (Meta)data uses a formal, accessible, shared, and widely applicable language for knowledge representation (e.g., the dataset is in JSON-LD);
- The (meta)data uses vocabularies that follow FAIR principles;
- (meta)data includes qualified references to other (meta)data (e.g., the metadata of a dataset can reference other similar datasets).

Reusability, i.e., data sets must be well formatted so that their use can be replicated in different situations:

- The (meta)data is richly described with a plurality of accurate and relevant attributes (e.g., in addition to the data having self-explanatory attributes, information such as “For what purpose was the data generated/collected?”, possible biases, whether the data is raw or processed, should be specified);
- The (meta)data is released with a clear and accessible data use license (e.g., the dataset is licensed by an MIT license);
- (Meta)data are associated with detailed provenance (e.g., the metadata contains a page describing the history/origin of the dataset);
- The (meta)data meets domain-relevant community standards (e.g., datasets must be formatted in a standardized way, such as JSON-LD, to allow for reusability).

If you do not use a ready-made dataset, it will be your responsibility to ensure that the dataset generated to train your model follows these criteria for good behavior. Much of machine learning engineering comes down to building datasets. So, during this process, it is important to make the description of the data types and attributes being

collected/used as clear and detailed as possible. Here are some extra recommendations:

- Catalog the number of samples, for each sensitive attribute, that your set has (e.g., how many samples are male, how many are female, how many samples do not have the gender attribute declared or declare themselves non-binary. Does your dataset allow the entire gender spectrum to be represented?);
- Describe the source domain of your data (e.g., were they voluntarily provided? Is "web crawling" for commercial purposes allowed in your country? How might the creation of your dataset conflict with the Privacy principle?);
- Know the dataset intimately, as it will be your responsibility to identify potential biases before your deployment phase. Remember that not all biases are bad, but certain types of biases can generate unwanted consequences;
- Share your findings. If we want to develop transparent systems, open-source projects should be the standard for the AI industry.

Other tools, such as FAIR, can be found in the literature. Gebru et al. (2018) provide a similar tool to evaluate datasets used for machine learning. In any case, and regardless of the tool used, it is important that datasets are documented/analyzed such as to prevent unwanted consequences from occurring after such models are deployed in the real world.

Reporting the results of a safety and ethics analysis is another important step for developers. In the same way that drugs are sold with package leaflets containing contraindications, dosages, and side effects, machine learning models must also be presented transparently.

Digital Catapult AI Ethics Framework

Developed by the Digital Catapult Ethics Committee,[10] he Digital Catapult AI Ethics Framework is an interview/questionnaire methodology. The questionnaires cover seven concepts, where each concept is explored by specific questions. These questions aim to explore how an organization is implementing ethical concerns in its product development.

The idea behind the Digital Catapult AI Ethics Framework is that by going through this questionnaire, possible security flaws or certain types of misconduct will be better explained, and developers made aware of their existence. The concepts worked on by this methodology, as some examples of its questions are:[11]

Clear benefits: The benefits offered by a product must be clear and transparent. At the same time, the benefits should outweigh the potential risks associated with the product developed.

- What are the goals, purposes, and intended applications of the developed product?
- Who or what can benefit from the product? Consider all potential beneficiary groups, whether individual users, groups, or society and the environment as a whole.
- Could these benefits change over time?

[10] The Digital Catapult Ethics committee seeks to translate theory in AI Ethics into practice. The committee is chaired by Luciano Floridi, Professor of Philosophy and Information Ethics at Oxford University, and director of the Digital Ethics Lab at Oxford University. https://migarage.digicatapult.org.uk/ethics/ethics-committee/.

[11] This tool, in its full version, can be accessed at the following address: https://migarage.digicatapult.org.uk/wp-content/uploads/2021/07/DC_AI_Ethics_Framework-2021.pdf.

Know and manage risks: The possible risks associated with the improper or intended use of the product should, as far as possible, be known by the developers.

- Have the risks of other foreseeable uses of the product, including accidental or malicious misuse of the product, been considered?
- How can potential risks or perceived risks be communicated to users, potentially affected parties, purchasers, or commissioners?
- Have all potential groups at risk, whether individual users, groups, or society and the environment as a whole, been considered?

Use data responsibly: compliance with current legislation (e.g., LGPD - Law No. 13.709/2018),[12] as well as other tools that help ensure that data is collected and used ethically (e.g., FAIR), are a basic starting point for any ethical assessment.

- How was the data obtained and how was consent obtained? Is the data current?
- Have potential biases contained in the data been examined, well understood, and documented? Is there a plan in place to mitigate them?
- Can individuals remove themselves from the data set? Can such people also remove themselves from any resulting model?

Be trustworthy: the burden of proof that your product is reliable and competent must be properly supported and proven by the developers. This burden must also be delivered in an interpretable format so that users are not misled or confused.

[12] Available at: http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709compilado.htm#art65.

- Within the company, are there sufficient processes and tools in place to ensure transparency, accountability, reliability, and appropriateness of the product developed?
- Is the nature of the product communicated in a way that intended users, third parties, and the general public can access and understand?
- Who is responsible if things go wrong? Are these the right people? Are these people equipped with the necessary skills and knowledge to assume such responsibility?

Diversity, equality, and inclusion: in a plural and diverse world, organizations that value principles such as diversity, equality, and inclusion should be the model to be followed. Thus, organizations must be able to foresee the possible consequences of the implementation of a product for a wide range of groups, with the intention that their product will be able to mitigate the inequalities and injustices that are structured in our political, cultural, and social fabric.

- Are there processes in place to establish whether the product may harm the rights and freedoms of individuals or groups?
- Does the organization have a policy on diversity and inclusiveness concerning recruiting and retaining staff?
- Where do ethics fit into the company's hiring practices? For example, are ethical questions raised in interviews?

Transparent communication: communication between developers (and the organization in general) and users, potentially affected parties, investors, and commissioners, should be transparent, clear, and intelligible. At the same time, the communication paths between these groups should allow concerns and complaints to be addressed efficiently.

- Does the organization communicate directly, clearly, and honestly about any potential risks of the product being supplied?
- Does the company have a clear, easy-to-use system for third-party/user or stakeholder concerns to be raised and addressed?

- Is there a communication strategy or process in place if something goes wrong (e.g., request for return, recall)?

Business model: the concept of "fair dealing" should be an integral part of a company's organizational culture, so that blind maximization of capital is not the only "normative guide" that guides and drives such an organization. In other words, ethical organizations should also be driven by maximizing "Social Good."

- Is environmental impact considered when choosing suppliers? Have options with clean energy sources been considered?
- Has differential pricing been considered? Are there any ethical considerations regarding pricing strategy?
- Are there any vulnerable groups that might receive lower prices?
- Is there data that third parties (e.g., charities, researchers) could use for public benefit?

The idea behind an interview conducted via the Digital Catapult AI Ethics Framework is that neglected problems and facts are brought into the light of the debate, and thus mitigation measures can begin to be planned and devised.

VCIO (Values, Criteria, Indicators, Observables)

Krafft et al. (2020), through the AI Ethics Impact Group (led by the VDE Association for Electrical, Electronic & Information Technologies, and the Bertelsmann Stiftung), proposes another type of translational tool. The authors present the VCIO model, something that, according to the authors, is a unique approach in the field of AI Ethics (Krafft et al., 2020, p. 6).

Like the Digital Catapult AI Ethics Framework, the VCIO model is a way to contextualize ethical concerns within the scope of application of a given model. The VCIO model is a multi-methodological framework, where AI systems are: (1) evaluated against a series of pre-established ethical principles; (2) results are distilled into an ethics label (AI Ethics Label); and finally (3), applications of the model are ranked using a risk matrix.

VCIO is an approach that seeks to identify observable indicators that can serve as decision criteria to determine whether an ethical principle is being preserved or not. This approach also seeks to clarify when conflicts between different values exist. For example, in certain applications (e.g., medical research), there is a trade-off between transparency and privacy where it is almost impossible to satisfy both sides (i.e., full transparency may come to mean little privacy and vice versa).

Thus, the VCIO approach operates on four levels:

- *Values:* that which should guide our actions;
- *Criteria:* that which defines whether a value (e.g., Justice) has been violated or not;
- *Indicators:* since criteria (as well as values) cannot be directly observed, we need indicators that can signal whether criteria are being met;
- *Observable:* aspects that can be observed and monitored by indicators.

According to Krafft et al. (2020, p. 16):

> However, it is not possible to derive the lower levels [Indicators and Observables] from the higher ones [Values and Criteria] in a direct, i.e., deductive way. Instead, the normative load runs through all four levels and requires further deliberations at all levels, in the course of which particular instances must be negotiated in detail. [...] Since there are no deductive relationships between values, criteria, indicators, and observables [...] normative decisions must be made in a scientific and technically informed context.

As an example, if we determine "Predictive Parity" as a criterion for "Fairness," we can use the accuracy of a model as an indicator and monitor this performance metric concerning different groups (observable quantity). If we determine "Sustainability" as a value, we can use "carbon footprint" as a criterion, monitoring, for example, the carbon footprint generated to train a specific model. Or we can monitor whether a particular organization chooses clean energy sources to train its models and run its servers.

Since there is no clear and objective way to determine criteria, indicators, and observables of the chosen values (we can even say that the choice of all these will be a normative choice by nature), the burden of proving that there is a correlation between what is advocated and what is monitored falls on the developers.

If there are (and usually there are) conflicts between values, developers can rank them to prioritize (depending on the context in which a model will be applied) different values. For example, developers may choose to prioritize values whose criteria, indicators, and observables are clearer to monitor and quantify. If a value has no clear way of being monitored (e.g., Accountability), it can be used as a tie-breaker between two conflicting values (e.g., between breach in privacy or lack of transparency, for which

of these violations will be easier to hold those responsible accountable? Which violation is likely to cause the most harm to those involved?).

In the table below, we see the application of the VCIO model in the analysis of "Justice":

Value	Justice		
Criteria	Evaluation of different sources of possible biases to ensure Fairness/Justice.		
Indicators	Was the training data analyzed to identify possible biases?	Have data labeling procedures been evaluated?	Does the model have predictive parity across different demographic groups?
Observables	Yes, all potential model biases were reported.	Yes, data labeling has been inspected by external reviewers.	Yes, predictive parity is guaranteed.
	Only a few biases are known to the developers.	Yes, data labeling has been inspected by internal reviewers.	Predictive parity is guaranteed only within a predetermined error percentage.
	No.	No.	No.

This table has been adapted and modified from one of the examples provided by Krafft et al. (2020, p. 22).

Tables like this can contemplate several different values, where for each value we can assign more than one criterion, each with its respective indicators and observables (the table above is just a simplified example). Thus, the main idea of the VCIO model is to rank values, criteria, indicators, and observables, so that abstract concepts (e.g., Justice) can be anchored in observable variables (e.g., accuracy rates are equivalent

for all groups considered by the model within a pre-established acceptable error limit).

To facilitate the interpretability of the analysis proposed by the VCIO model the results are then condensed into an "Ethics Label," i.e., an indicator that is easy to understand for citizens, users, consumers, legislators, or regulatory bodies.

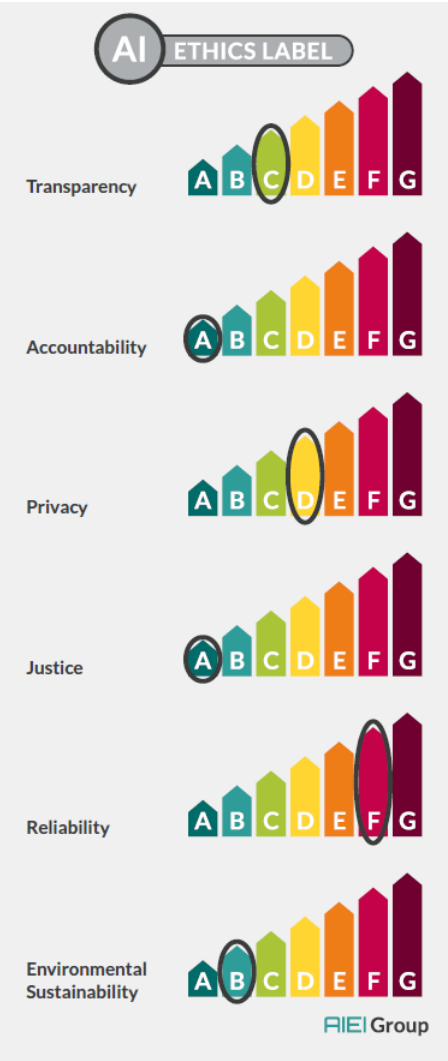
The label proposed by Krafft et al. (2020) includes a rating for each of the values contemplated by the VCIO ethical analysis. In the example below, the values used are *Transparency*, *Accountability*, *Privacy*, *Fairness*, *Reliability*, and *Sustainability*. However, the model can certainly be extended to contemplate other values.

The suggested ranking is done by letters, from A to G (seven levels), where "A" is the highest-ranking (e.g., a model with an "A" score in Justice is a model where all, or most, of the criteria are met by the most stringent observable measures). Krafft et al. (2020) suggest seven levels so that the granularity of observables is better expressed (in the table above, we use only three).

If we chose to use the example table in this Guide, we could choose to create a rating with only three levels ("A," "B," and "C," or "Green," "Yellow," "Red"). Each observable would correspond to a rating (e.g., "Green = A"), and the final grade assigned to an AI system would be made by aggregating the different observable ratings. For example, a model

might receive a “B” rating in Justice if it contains two indicators with “A” observables and one indicator with a “C” observable.[13]

After the rating levels are defined, as well as the granularity of the observables, a way to aggregate such scores still needs to be defined. There are several ways to do such an aggregation procedure, and Krafft et al. (2020) suggest methods such as arithmetic average, harmonic average, and even the definition of minimum criteria for certain ratings to be achieved.[14]



AI Ethics Label (Krafft et al., 2020, p. 13)

Another step we need to take in an ethical analysis is to evaluate the context and the potential associated risks of an application. As stated, there is little (if any) ethical analysis required when implementing an AI model created for industrial process monitoring.[15] However, there are contexts where AI systems, from an ethical standpoint, should never be used. For example, a high-risk decision, such as whether or not to turn off life support equipment of a brain-dead patient, should not (in principle) be made by a statistical inference model. Or, capital punishment (i.e., the “death penalty”) should never be prescribed and sentenced by an AI system.

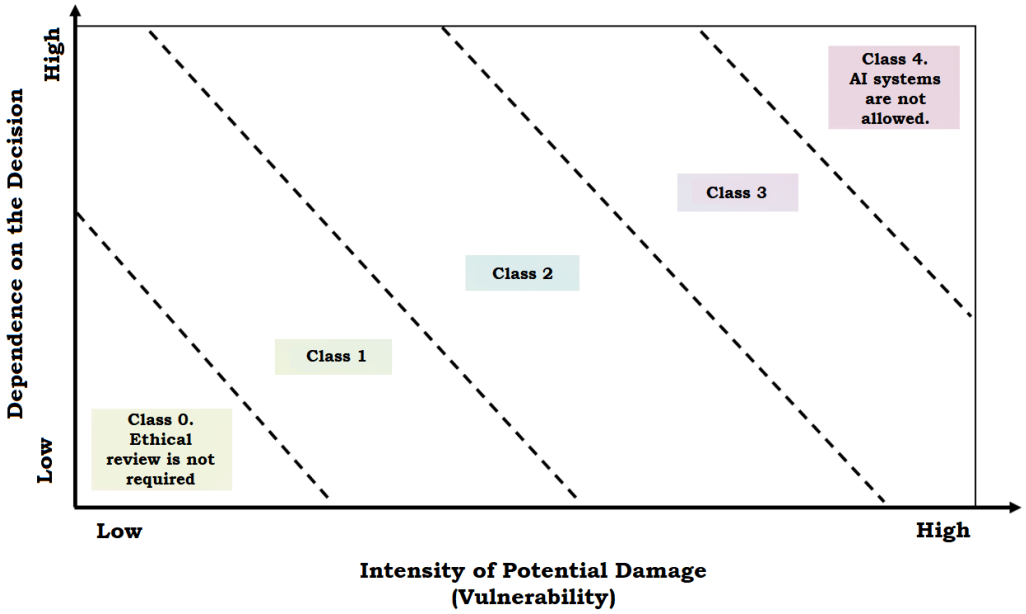
[13] All model biases are known/explained (A); the model guarantees predictive parity (A); however, the data labeling procedure used for training the model has not been audited, either by internal or external assessors (C) (i.e., “A + A + C = B”).

[14] The details and nuances of the VCIO model can all be found in the original publication by Krafft et al. (2020).

[15] It may be necessary if such a model will cause some labor displacement, i.e., people will lose their jobs to AI systems.

Thus, when we analyze the context of an application it can also tell us the rigor of our evaluation (e.g., which observables must be met for a model to receive a minimum rating), whether our evaluation is necessary (e.g., the model does not require any ethical review), or whether the model should not be deployed at all (e.g., the result of the ethical review recommends not allowing the deployment of a certain application).

Krafft and Zweig (2019) suggest that the risks of a given application be analyzed in a two-dimensional risk matrix. Risk matrices are commonly used for evaluating the risk level of a system. The matrix proposed by Krafft and Zweig (2019) has two factors: (1) the intensity of the potential damage; and (2) the dependence of the person(s) affected by the respective model. The authors divide their risk matrix into five classes, from 0 ("no risk") to 4 ("model deployment should not be allowed").[16]



Risk Matrix (Krafft & Zweig, 2019, p. 32).

[16] According to Krafft & Zweig (2019), the risk analysis model is not designed to classify all possible risk contexts exhaustively. We can certainly increase the granularity of the risk spectrum. however, the main idea behind this tool is still meaningful, i.e., that the risk of a given application should be assessed before its deployment.

In this matrix, the risk is defined by two different axes. The vertical axis represents how much the decisions of a model (ADM - “algorithmic decision-making systems”) could affect people. Below are some questions that can guide us in evaluating this dimension:

- *What function is the system automating (e.g., discerning cats from dogs, or determining the emergency shutdown of a nuclear power plant)?*
- *Is there any human supervision?*
- *If the system malfunctions, how might this affect the parties involved?*
- *Can the system's decisions be contested? In what way?*

While the horizontal axis represents the intensity of the potential damage caused by the model's outputs:

- *Could the model impact fundamental human rights?*
- *What is the level of this impact (e.g., loss of a benefit, physical harm, loss of a life)?*
- *Will the impacts be on individuals? Legal entities? Individuals or Organizations?*

As examples of AI models for each class, we can mention:

- *Class 0:* systems for automation of industrial processes, systems for automation of weather forecasts, systems for product recommendation;
- *Class 1:* recommendation systems for personalized searches on search engines, recommendation systems on social networks, recommendation systems on streaming platforms;
- *Class 2:* personalized recommendation systems for jobs, personalized recommendation systems for services, language models for conversation (i.e., chatbots);
- *Class 3:* recommendation systems for election advertisements, computer vision systems for law enforcement, criminal recidivism

assessment systems, credit score assessment systems, autonomous vehicles;

- *Class 4:* autonomous weapons, autonomous judges.

Another example of a risk matrix is the MIL-STD-882E risk matrix (Military Standard 882, Department of Defense Standard Practice System Safety, USA).[17] The MIL-STD-882E risk matrix for qualitative assessments has two assessment categories: *Severity* and *Probability*.

Risk Evaluation Matrix				
Probability/ Severity	**Catastrophic (1)**	**Critic (2)**	**Marginal (3)**	**Negligible (4)**
Frequent (A)	**High**	**High**	**Serious**	**Average**
Probable (B)	**High**	**High**	**Serious**	**Average**
Occasional (C)	**High**	**Serious**	**Average**	**Low**
Remote (D)	**Serious**	**Average**	**Average**	**Low**
Improbable (E)	**Average**	**Average**	**Average**	**Low**
Eliminated (F)	**Eliminated**			

Severity can be defined by the following set of categories:

- *Catastrophic:* risk of death (e.g., autonomous weapons attacking civilians);

[17] MIL-STD-882E, Department of Defense Standard Practice: System Safety (May 11, 2012). http://everyspec.com/MIL-STD/MIL-STD-0800-0899/MIL-STD-882E_41682/.

- *Critical:* risk of serious injury (e.g., traffic accidents caused by autonomous vehicles);
- *Marginal:* minor damage/injury (e.g., incorrect/harmful classifications generated by an ADM);
- *Negligible:* negligible damage/injury (e.g., your video feed does not contain your favorite series).

Meanwhile, "Probability" is the estimation of the frequency of an event that might happen in the future (something that is often difficult, or impossible, to determine precisely):

- *Frequent:* event that can occur frequently (e.g., one misclassification every 10 samples);
- *Probable:* will occur several times in the life of the system (e.g., one misclassification every 100 samples);
- *Occasional:* events that may occur at some point in the life of the system (e.g., one misclassification every 1,000 samples);
- *Remote:* event unlikely to occur, but may still occur (e.g., one misclassification every 10,000);
- *Improbable:* event extremely unlikely to occur (e.g., one misclassification every 100,000);
- *Impossible:* Equals a probability of zero.

Certainly, the examples cited above can be challenged. Given the ambiguous and context-dependent nature of Ethics/Safety when applied to complex real-world situations, arguments can be made as to which class an application "really" belongs to or what is the "true" level of severity/probability of an AI system failing to act safely. [18] However, I believe that what is important is not exactly the result (i.e., the "exact" classification of an application) but rather the deliberation process that will lead to that result (i.e., the ethical analysis itself).

[18] One wrong classification for every 1,000 samples may not seem like much, but if the application being evaluated makes one call per second to the model, and the model operates for 4 hours/day, that is 14400 calls to the model per day (~15 errors per day). Depending on the application, this could be considered high risk.

At the same time, to optimize the regulatory processes of AI systems, a risk analysis (e.g., by a division of risk classes) can help define how rigorous our assessment should be. In this way, applications that fall into different risk classes should be approached differently.

For example, we may come (as a society) to define that while applications involving low risk (e.g., classes 0, 1 and 2, or Eliminated, Low and Medium risk levels) may be audited internally (i.e., by the organization itself), high-risk applications (e.g., classes 3 and 4, or Serious and High) must also be audited by external regulatory bodies (e.g., the government, the ACM, the IEEE). We can also define that for AI systems to be safely deployed, certain application classes must obtain minimum values in their evaluation (e.g., all applications that fall into Class 2 must obtain a "B" evaluation in all evaluated values).

Morley et al. (2021, p. 250) summarize the concept of "Ethics as a Service" into two types of "spheres of accountability," which synthesize the concerns raised by the translational tools presented in this section:

- *Internal Accountability:* Define contextually the meaning of each ethical principle spelled out by a Code of Ethics created by regulatory bodies (i.e., external accountability). Select the use of tools/methods from a pre-approved list of available tools/methods. Conduct an ethical review of the product itself at all stages of development and implementation, including a forward-looking ethical review for the future.
- *External Responsibility:* Develop a Code of Ethics, review it regularly, and develop a process that AI developers should follow to contextually apply the ethical principles defined by such a code. Evaluate available tools/methods, and compile a pre-approved list of such tools for developers to use in developing their products. Audit the AI systems developed to ascertain their compliance to the

current Code of Ethics (e.g., IEEE's Ethically Aligned Design; Bill 21/2020;[19] ACM's Code of Ethics[20]).

Distributing the accountability of AI governance (and its Ethics operationalization) in this way ensures a relatively clear way of what the roles of the different actors in this hierarchy of services are. Whether it is a member of the IEEE ethics committee working towards updating the current Code of Ethics or a safety engineer at a company performing a diagnostic/evaluation of a model, each actor has their role to play.

In the end, the translational tools presented can be used individually or together. They provide a general approach to implementing ethics in the development of intelligent systems:

- An organization planning to develop an AI system for a specific application may use a questionnaire/checklist (e.g., Digital Catapult AI Ethics Framework) to determine the ethical risk of an application. Depending on the risk involved (e.g., VCIO model risk matrix classifies the application as “Class 0”), the process ends at this stage. If there are ethical issues to consider, then the organization performs a full assessment of its application;
- A complete evaluation should cover all the development stages of an AI system (i.e., Data Collection, Model Development, Model Evaluation, Post-processing, Deployment, Monitoring). Each stage can be compromised by different sources of problems (e.g., collected data is biased by historical biases). Tools like FAIR, the Digital Catapult AI Ethics Framework, the VCIO Model, among others, can help developers make such problems more evident;
- Each application context has its own specificities. Certain ethical values and principles may not make sense for a given application.

[19] Bill that establishes the foundations, principles and guidelines for the development and application of artificial intelligence in Brazil. https://www.camara.leg.br/proposicoesWeb/prop_mostrarintegra?codteor=1853928.

[20] ACM (Association for Computing Machinery) Code of Ethics and Professional Conduct. https://www.acm.org/binaries/content/assets/about/acm-code-of-ethics-and-professional-conduct.pdf.

It is up to developers (and regulatory bodies) to assess which ethical principles should be prioritized in a given application context. Ethical principles should be grounded in observable and verifiable quantities so that an ethical evaluation can be based on certain objective evaluation criteria (e.g., the VCIO Model seeks to accomplish this with the "Values, Criteria, Indicators and Observables" methodology);

- Depending on the context, certain values are found in opposition (e.g., Transparency and Privacy) while other values may only be approximated within a context of applicability (e.g., "total" algorithmic justice suffers from an impossibility theorem). It is up to the developers to make the compromises and commitments made explicit in a transparent way;
- This assessment process, depending on the risk involved by the application (e.g., VCIO model risk matrix, MIL-STD-882 E risk matrix) can be directly audited by the organization or require an external assessment, performed by the responsible regulatory bodies;
- After the end of an evaluation, the results must be presented clearly and transparently to all parties involved in the use of the developed system (e.g., VCIO Model Ethics Seal).

However, we must remember that certain problems may only arise after the implementation phase of a model. So, we cannot reduce ethical analysis and safety engineering to just *"checklists to be filled in."*

Are translational tools enough?

As much as translational tools help bring ethical theory closer to the practice of intelligent systems development, we must be aware that such strategies alone do not guarantee that a given model/product will not generate unintended consequences.

The actors responsible for administering an ethical evaluation, with their particular ethical notions, may not always be aligned with the "Social Good" (Green, 2019; Krishnan, 2019). Thus, there needs to be an effort to align such views. In other words, developers must have an understanding of what "Social Good" means. This is why ethical review committees should always be formed by an interdisciplinary group with members from various fields of knowledge (e.g., engineers, computer scientists, philosophers, sociologists, lawyers, etc.).

One criticism raised against translational tools is that such methods are "extra-empirical" (Fazelpour & Lipton, 2020). That is, while such tools seek an empirical and objective basis for testing and evaluating notions of ethics in the development of intelligent systems, these tools themselves are not "per se" subject to "empirical and objective" evaluation. Something that, as Morley et al. (2021) point out, makes such methodologies subject to manipulation by those applying them.

An ethics assessment cannot be reduced to just a "one-off" test or an inventory to be filled out. The role of the safety engineer in AI ethics is a constant process, as models must be constantly monitored. Without constant maintenance of these models, translational tools do not guarantee that an AI system will be beneficial or safe. Imagine an elevator company where no periodic routine evaluation and inspection of their products is performed, and they only sell elevators with a label saying "100% safe". Would you buy (or use) one of this company's elevators?

You couldn't even (legally) buy such a product because, in most countries, companies that do not implement a “Preventive Maintenance Program” for this kind of technology cannot even legally provide services.[21]

Just as this kind of implementation is already a “standard” procedure for technologies such as elevators, the same must become routine for the maintenance of AI systems. Intelligent systems cannot be produced, implemented, and then abandoned by their developers. And that is what is expected of an organization that truly seeks to develop ethical and safe artificial intelligence.

For this to be achieved, Ethics cannot be reduced to diagnostic and evaluation procedures but must be treated as a preventive service that must be regularly employed.

Starting in the next section, we will see how security issues have been addressed by the literature and the private sector, and how we can augment the qualitative methodologies presented so far with more quantitative tools.

[21] SIT Ordinance No. 224 of May 6, 2011. https://www.legisweb.com.br/legislacao/?id=232119.

AI Safety

AI Safety is in itself its own research area, with its own concerns. This area arose from the need to develop methods to deal with systems that are opaque, complex, fragile when operating outside their distribution, not modulable, and difficult to interpret. And such systems need their own special form of treatment:

> Just as, historically, security methodologies developed for electromechanical hardware have not generalized well to the new issues raised by software, we should expect that software security methodologies will not generalize well to the new complexities and dangers of Machine Learning (Hendrycks et al., 2021a, p. 2).

Jurić et al. (2020), in their bibliometric review of the literature in AI security, suggest that the main topics being worked on in the area are:

- *Interpretability:* How to interpret the decision-making of opaque algorithms, such as deep neural networks (Guidotti et al., 2018)? At the same time, how to interpret the results of our own interpretability tools?
- *Corrigibility:* How to make potentially flawed agents, even if rational agents (expected utility maximizers) have a strong instrumental incentive to preserve their terminal goals, correctable (Soares et al., 2015)?
- *Robustness to Adversarial Attacks:* Neural networks are highly susceptible to adversarial attacks, i.e., attacks specially designed to trick them (Yuan et al., 2019). How can we protect our systems against these forms of attacks?
- *Safe Exploitation and Distributional shift:* Generally, the training domain is not a perfect representation of the real domain where the agent will operate. How can we ensure the “safe” behavior of our

models when operating in domains very different from those seen in their training (Amodei et al., 2016)?

- *Value Learning and Goal Specification:* As we seek to integrate AI systems into increasingly complex environments, the tasks we expect such systems to solve also become more complex. Specifying an objective function to be optimized by an AI system in a "clear" way (i.e., without specification errors) is not a simple task, as human values and preferences can be extremely difficult to specify (Soares, 2016).

Meanwhile, Hendrycks et al. (2021a) present the following technical problems that we encounter in machine learning. These problems tend to become gradually more prominent as models are implemented in increasingly complex and high-risk applications:

- *Robustness:* The creation of models that are resilient to adversarial attacks and unusual situations (i.e., situations outside their training distribution). Currently, models trained by machine learning are still fragile and rigid, not operating well in dynamic and changing environments. In a world full of rare events happening all the time, such models must be extremely robust;
- *Monitoring:* The detection of malicious use, malfunction, or unintended functionality. Just as nuclear power plants are monitored by HROs (high-reliability organizations), future machine learning systems may be monitored in the same way (e.g., intelligent traffic management systems controlling cities populated by autonomous cars). Thus, it becomes necessary to develop methodologies to aid the monitoring and supervision of such systems;
- *Alignment:* Creating models that robustly optimize hard-to-specify goals (e.g., human values). AI systems often exhibit a certain level

of agency (e.g., they possess and optimize goals). Something that differs such systems from other forms of technology. Ideally, we would like to create agents that "prefer" good world-states. However, what defines a "good world-state"? Goal proxies can be: (1) difficult to specify; (2) difficult to optimize; (3) fragile; and (4) stimulate unwanted behavior (e.g., reward hacking);

- *External Safety:* Models can be embedded in insecure environments, such as malfunctioning software and poorly structured organizations. Given the fragility that models trained by machine learning exhibit, it is important to make their deployment environments secure, either by developing software resilient to cyberattacks or by creating governance policies aimed at making the deployment of such models secure.

It is important to note that all of the cited avenues of research, with their particular problematics, remain open problems in AI Safety (and of Machine learning itself).[22]

Like any emerging research field, the concerns and contributions coming from AI Safety have not yet penetrated the "mainstream" of the industry and academia. For example, if we go through the major advanced AI research and development (R&D) projects (i.e., projects that seek to advance the state-of-the-art of the field), we see that only a small minority conduct any kind of safety-focused research.

In 2017, Baum (2017) identified 45 R&D projects with the goals of developing advanced AI. Of the 45 projects reviewed, only 13 had active involvement with the area of security, while the vast majority did not specify any type of research focused on the area of AI Safety. Fitzgerald et al. (2020) updated Baum's (2017) findings, increasing the project count

[22] For those interested, Critch & Krueger (2020) present an extensive analysis, with several suggestions and avenues for research, of the AI Safety field.

to 72 active 2020 R&D projects focused on developing advanced AI. Of the 72 projects listed, only 18 have active engagement with AI safety.

We have produced a table/summary of the findings from Fitzgerald et al. (2020), "2020 Survey of Artificial General Intelligence Projects for Ethics, Risk, and Policy," a paper commissioned by the Global Catastrophic Risk Institute. In this table are: (1) the name of the Project (with link to its webpage); (2) the country/leader hosting it; (3) the institution (and type of institution) responsible for the project; (4) whether such project has ties to the Military sector; (5) whether the project is Open Source; (6) the size of the project; and finally (7) the engagement with the AI Security area of each project. The table can be found at the link cited in the footer.[23]

From these results, we can see that, as stated at the beginning of this section, AI Safety is still "something new to be integrated." However, we have good examples of organizations that invest and care about the ethical and secure development of their applications. Let's take two of the largest organizations involved in AI development as an example: DeepMind[24] and OpenAI.[25]

DeepMind is a Google project based in London (UK) led by Demis Hassabis and Shane Legg. From their labs, in addition to some of the most proficient and general AI models ever produced (Mnhi et al., 2013; Silver et al., 2016; Badia et al., 2020), many AI Safety-related studies have been produced and published (Leike et al., 2017; Everitt et al., 2019;

[23] AI Safety Watch: Advanced Artificial Intelligence R&D (2020). https://en.airespucrs.org/post/ai-safety-watch-advanced-artificial-intelligence-r-d-2020.

[24] https://deepmind.com/.

[25] https://openai.com/.

Mikulik et al., 2020; Kenton et al., 2021). DeepMind also collaborates with OpenAI on projects focused on AI Safety.

OpenAI, meanwhile, a non-profit AI research organization, is also responsible for pushing the state-of-the-art in several areas of the field (Brown et al., 2020; Chen et al., 2021), publishing most of its findings open-source, and openly promoting its mission to *"try to directly build a safe and beneficial AGI."* [26]

Let's take as an example two of the most recent models released by OpenAI: GPT-3 and Codex. [27]

GPT-3 (Generative Pre-Train Transformer 3), a Transformer with 175 billion parameters, is a machine learning model trained in an unsupervised way (Self-Supervised) capable of generating samples of texts such as poems, articles, news, as well as solving several problems linked to NLP, without requiring any post-processing or tuning. However, what kind of unwanted behavior can we expect from such a model interacting with the real world?

In their publication, Brown et al. (2020) conduct an extensive safety analysis of the developed model. In it, the authors report on potential malicious applications (e.g., misinformation, spam, cybercrime), issues related to equity, bias, and representativeness (e.g., gender, race, religion), and even energy consumption related to the use of the model (i.e., Sustainability).

Codex, on the other hand, is a model capable of transcompiling commands given in natural language into code (e.g., Python). Codex has been trained from GPT language models tuned to open-source public

[26] https://openai.com/about/.

[27] These models have not yet (for security reasons) been openly released to the public. However, the publications by Brow et al. (2020) and Chen et al. (2021) describe the process of training such models. The models can also be accessed via API through the OpenAI beta platform, available at: https://beta.openai.com/.

code repositories (GitHub). Let's look at an example generated by the OpenAI beta API:

Command:

"""

1. Create a function that counts prime numbers between 0 and 100.
2. Plot the number of prime numbers between 0 and 100 as a step function.
3. Label the plot "Number of Primes between 1 and 100."
4. Use sympy, numpy, and matplot as modules.

"""

Output:

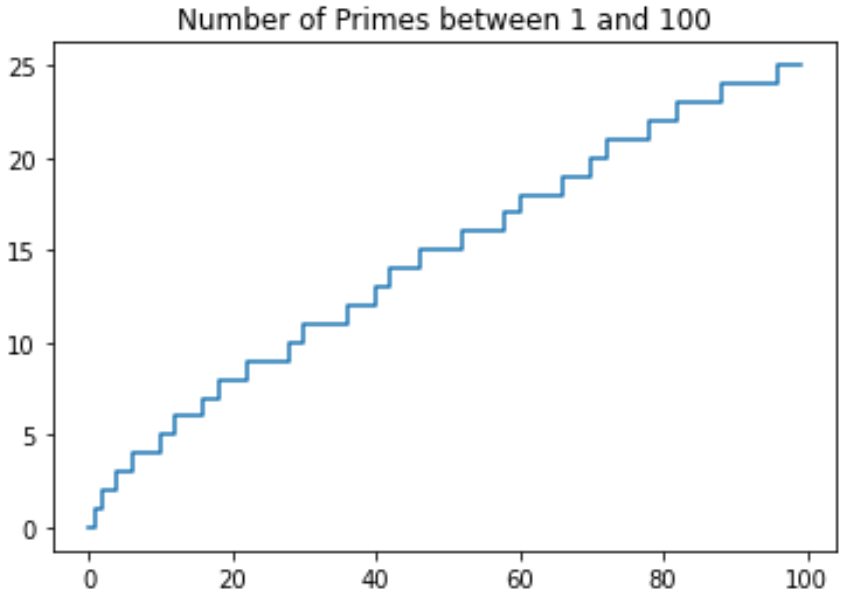
```
import sympy
import numpy as np
import matplotlib.pyplot as plt

def prime_count(n):
    count = 0
    for i in range(2, n+1):
        if sympy.isprime(i):
            count += 1
    return count

x = np.arange(0, 100, 1)
y = np.array([prime_count(i) for i in x])

plt.step(x, y)
plt.title("Number of Primes between 1 and 100")
plt.show()
```

You can test this result yourself. Just copy and paste the above code into a Google Colab notebook and press "play." The program will produce what was requested (and gets the number right, i.e., 25):



Again, we can see in the publication by Chen et al. (2021) an extensive analysis of the implications and possible impacts of this type of technology, something that promises to make the ability to write and generate code extremely accessible to anyone who can read and write English language commands (or is literate and has access to a translator).

In their publication, Chen et al. (2021) mention risk factors such as:

- Codex does not always produce code that is aligned with the programmer's intent. Here, we define misalignment (or “alignment failure”) as “when the system is assigned to perform some task X, and the system is capable of performing X but 'chooses' not to do so” (Chen et al., 2021, p. 11). This contrasts with the situation where a system does not do X because it cannot do X, i.e., the system is just incompetent;
- Codex may suggest solutions that superficially appear correct but which do not accomplish the task intended by the user (i.e., overconfidence);
- As in the case of other language models, Codex may be asked to produce code/comments that contain racist and denigrating content;

- The authors evaluated the economic impacts on the labor market that automatic code generation models may cause (e.g., reducing the value of the work of software engineers and developers);
- The authors assessed the likelihood that automatic code generation models assist in the creation of malware (assisting in the realization of cybercrimes);
- The authors evaluated the environmental impact of training and using large models such as Codex-12B (GPT-tuned code generation with 12 billion parameters). For example, it is estimated that training GPT-3 produced about 552 metric tons of carbon dioxide, equivalent to what more than 120 cars produce in a year; [28]
- The authors evaluated how likely it was that the trained model would generate code identical to code found in public repositories (GitHub). Something that could have unwanted legal implications (i.e., violation of private property rights).

Given all the possible documented risks, the authors further state that:

> [...] given the above, models such as Codex must be developed, utilized, and their capabilities carefully explored to maximize their positive social impacts and minimize the intentional or unintentional harm that their use may cause. A contextual approach is fundamental to effective risk analysis and mitigation, although some categories of mitigations are important to consider in any deployment of code generation models (Chen et al., 2021, p. 13).

This is a good example of a product that has been developed under a robust ethics and safety regime. Robust in the sense that the problems and limitations of the model created are (as far as possible) known to the

[28] However, as much as training large models like GPT-3 requires large amounts of energy, its inference in, for example, generating 100 pages of content, can cost in the order of 0.4 kW/h.

developers, who have in turn taken the initiative to report them to the interested community. [29]

This is one of the roles of the AI safety engineer. Not only to evaluate the possible biases and problems that may arise during the training of a model and after its implementation in a given context but to *seek to mitigate new problems that may arise.*

Not all potential uses of a model are always known to its developers. Perhaps early machine learning models had clear limits of use (e.g., classifying images of digits). However, the same is not true for models being generated today. Often models are capable of performing tasks far beyond those that their developers had in mind. Again, citing the model trained by OpenAI, GPT-3 was only trained to "predict the next word in a sequence." It was expected that the model would be proficient in NLP-related tasks. What was not expected was that the model would have "learned" arithmetic without explicit supervision.

To avoid being caught off guard, safety analyses must go a bit beyond the translational tools we have reviewed. We need quantitative methods to evaluate, stress, and attack our models. But how can we implement this kind of practice in the development of intelligent systems? In the next section, we will look at a tool for this task.

[29] As another example, we can cite Redwood Research, an organization that performs applied alignment research in AI. In 2021, the organization was developing techniques to control text-generating models (e.g., GPT-3) to prevent such models from producing text with unwanted content (the goal of the model being trained by the project was to detect when a text contained some kind of violence). More information at: https://www.alignmentforum.org/posts/k7oxdbNaGATZbtEg3/redwood-research-s-current-project.

The AI Robotics Ethics Society®

Safety Reports and Model Cards

Given that in certain contexts and applications, the use of AI systems must be robustly monitored. One way to connect the concerns and notes of developers with those who will use such models and applications involves creating documentation that details the performance characteristics of a given AI system, i.e., model cards.

We can define a model card as:

> [...] short documents accompanying models trained by machine learning that provide benchmarking under a variety of conditions, such as between different cultural, demographic, or phenotypic groups (e.g., race, geographic location, gender, skin type) and intersectional groups (e.g., age, gender) that are relevant to the intended application domains. Model cards also reveal the context in which the models are intended to be used, details of performance evaluation procedures, and other relevant information (Mitchell et al., 2019, p. 220).

We can think of a model letter as the result of a safety assessment of a given AI system. As much as there are no standardized and universal documentation templates yet, there are suggestions for what such templates should look like, and what kind of information should be explicit in a model letter (Bender & Friedman, 2018; Holland et al., 2018; Gebru et al., 2018).

A model card intends to provide users of a given system with information about:

- How to use the model;
- How *not to use* the model;
- The kinds of mistakes the model can make most often (i.e., its vulnerabilities).

Informed of this reality, users are expected to be able to use "imperfect models" in the best possible way. Model cards can also benefit many different types of actors:

- *AI developers* can better understand how well a model can work for an intended application, compare model results with other similar models, understand how a model can be improved, tuned, and combined with other models;
- *Software developers* who use predictions from AI systems can better design their applications;
- *Regulatory entities* can understand how an AI system may fail and impact people, and use such information to regulate the use of AI for certain high-risk applications;
- *People impacted by an AI system* can use a model letter to determine whether the impacts experienced were properly predicted and specified, and at the same time, know who is responsible for developing such a model/application.

We will draw on the work of Mitchell et al. (2019), "Model Cards for Model Reporting," to demonstrate how to use such a tool. In the authors' work, Mitchell et al. (2019) used two examples, an image classifier (i.e., a smiley detector) trained on the CelebA dataset, and a toxicity detection model (e.g., autonomous detection of texts with toxic content).

Model Card
Model Details (basic model information)
1. Organization/Individual who developed the model; 2. Date of development; 3. Model version (e.g., v 0.1); 4. Type of model (e.g., logistic regression model, convolutional neural network, transformer language model, vision transformer); 5. Information about training algorithms, parameters, features used, fairness constraints, or other approaches applied; 6. GitHub article/developer page/repository; 7. Information for citation; 8. License;

9. Where to submit questions and comments about the model.
Intended Use (use cases that were predicted during development)
1. Primary intended use (What is the intended use of this model?); 2. Primary intended users (What is the intended target audience of this model?); 3. Uses outside the intended distribution (What types of applications has the model not been trained to support?).
Factors (e.g., demographic groups, phenotypes, environmental conditions, technical assignments, or other relevant factors)
1. Relevant factors (What are the factors for which model performance may vary, and how were these determined?); 2. Evaluation factors (Which factors are being reported, and why were these chosen?).
Metrics (metrics should be chosen to reflect the potential real-world impacts of the model)
1. Model performance (e.g., accuracy, precision, recall, AUC, etc.); 2. Decision thresholds (If decision thresholds are used, what are they, and decision thresholds, what are they, and why were they chosen?) 3. Variance approaches (How was model variability measured? Standard deviation? Variance?).
Evaluation data (details of the dataset used for training and evaluating the model)
1. Dataset (Which dataset was used to evaluate the model?); 2. Motivation (Why was such a data set chosen?); 3. Preprocessing (How was the data preprocessed? Tokenization? Normalization? Were samples with “NaN” values excluded, or were their values estimated?) *4.* Training data (It is not always possible to provide such a set. When possible, this section should reflect the evaluation data. If such detail is not possible, minimal allowable information should be provided here, such as details of the distribution by various factors (e.g., distribution of subgroups across characteristics).
Ethical Considerations (an ethical review need not necessarily produce precise solutions, but the ethical contemplation process should be geared towards

informing stakeholders about concerns raised by developers and steps for future work)
1. Does the model use any sensitive data? 2. Is the model intended to inform decisions about issues central to human life? 3. What risk mitigation strategies were used during the development of the model? 4. What risks may be present in the use of the model?
Details and Recommendations (additional concerns not covered in the previous sections)
1. Do the results suggest any further testing? 2. Were there any relevant groups that were not represented in the evaluation dataset? 3. Are there any additional recommendations for the use of the model?
Quantitative Analysis (quantitative analyses should provide the results of the model evaluation according to the chosen metrics, broken down by the chosen factors)
1. Unit results (How did the model perform concerning each factor?); 2. Intersectional results (How did the model perform concerning the intersection of the factors evaluated?).

In the above card (Mitchell et al., 2019, p. 222), we see several types of information that can shed light on questions about the development, intended use, and potential problems of a given model. However, it is important to remember that the above list is not exhaustive or complete and that such reports should be sensitive to a development/application context.

For example, the amount of information that a private company is willing to make public (e.g., training data) may be less than an academic organization. Certain companies may choose not to disclose certain key information for the development of a commercial application (e.g., training algorithms). Even so, there are ways to present pertinent information (e.g., the performance of a model) without revealing confidential information (e.g., how such a model was developed).

In this work, we will use two different examples:

- A model for *credit card approval*, and;
- A model for *forecasting annual income*.

Through these examples, we will suggest some methodologies and tools to: (1) inspect a model trained by machine learning; and (2) "fill" a model card. However, it is important to remember that (by no means) the tools and methodologies presented in these examples are the entirety of AI Safety. Nevertheless, they can certainly assist developers in implementing an initial safety assessment.

Example 1: Credit Card Approval

Evaluation of credit card applications is a task that commercial banks commonly use artificial intelligence to automate. In this example, we will develop a logistic regression model (one of the most common techniques in machine learning) to solve a binary classification problem: classifying a credit card application (characterized with a series of features/features) as "Approved" or " Not Approved."

We will use the "Credit Approval Dataset" from the UCI Machine Learning Repository.[30] This dataset has 689 samples of credit card applications, labeled as approved or disapproved. However, to protect the privacy of the individuals in this dataset, all features have been masked, i.e., instead of using explicit feature labels (e.g., `Gender = ['Male, 'Female, 'Non-Binary]`), these values were replaced by symbols (e.g., `Gender = ['a', 'b', 'ab']`).

The features themselves have been removed. However, for this example, we will treat each sample as consisting of the following features (typically requested in credit card applications):

- `"Gender", "Age", "Debt", "Married", "Bank Client", "Education", "Race", "Years Employed", "Prior Default", "Employed", "Credit", "Driver's License", "Citizenship", "Postal Code", "Income";`

And as a target:

- `"Approval Status".`

[30] UCI Machine Learning Repository, Center for Machine Learning and Intelligent Systems. http://archive.ics.uci.edu/ml/datasets/credit+approval.

The data can initially be visualized as a Pandas[31] data frame:

	Gender	Age	Debt	Married	Bank Client	Education	Race	Years Employed	Prior Default	Employed	Credit	Driver's License	Citizenship	Postal Code	Income	Approval Status
0	a	58.67	4.460	u	g	q	h	3.04	t	t	6	f	g	00043	560	+
1	a	24.50	0.500	u	g	q	h	1.50	t	f	0	f	g	00280	824	+
2	b	27.83	1.540	u	g	w	v	3.75	t	t	5	t	g	00100	3	+
3	b	20.17	5.625	u	g	w	v	1.71	t	f	0	f	s	00120	0	+
4	b	32.08	4.000	u	g	m	v	2.50	t	f	0	t	g	00360	0	+
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
684	b	21.08	10.085	y	p	e	h	1.25	f	f	0	f	g	00260	0	-
685	a	22.67	0.750	u	g	c	v	2.00	f	t	2	t	g	00200	394	-
686	a	25.25	13.500	y	p	ff	ff	2.00	f	t	1	t	g	00200	1	-
687	b	17.92	0.205	u	g	aa	v	0.04	f	f	0	f	g	00280	750	-
688	b	35.00	3.375	u	g	c	h	8.29	f	f	0	t	g	00000	0	-

689 rows × 16 columns

As has been said, features (especially categorical ones) have been masked by "meaningless symbols." Functions like `.info( )` and `describe( )` can give us a more detailed overview of the data types we are working with.

We can quickly access how many subgroups each characteristic has (e.g., `Gender = 3, Education = 15, Race = 10, Prior Default = 2`) nd other important statistical data (e.g., mean, standard deviation, maximum values, minimum values).

	Gender	Age	Debt	Married	Bank Client	Education	Race	Years Employed	Prior Default	Employed	Credit	Driver's License	Citizenship	Postal Code	Income	Approval Status
count	689	689	689.000000	689	689	689	689	689.000000	689	689	689.000000	689	689	689	689.000000	689
unique	3	349	NaN	4	4	15	10	NaN	2	2	NaN	2	3	170	NaN	2
top	b	?	NaN	u	g	c	v	NaN	t	f	NaN	f	g	00000	NaN	-
freq	467	12	NaN	518	518	137	398	NaN	360	395	NaN	373	624	132	NaN	383
mean	NaN	NaN	4.765631	NaN	NaN	NaN	NaN	2.224819	NaN	NaN	2.402032	NaN	NaN	NaN	1018.862119	NaN
std	NaN	NaN	4.978470	NaN	NaN	NaN	NaN	3.348739	NaN	NaN	4.866180	NaN	NaN	NaN	5213.743149	NaN
min	NaN	NaN	0.000000	NaN	NaN	NaN	NaN	0.000000	NaN	NaN	0.000000	NaN	NaN	NaN	0.000000	NaN
25%	NaN	NaN	1.000000	NaN	NaN	NaN	NaN	0.165000	NaN	NaN	0.000000	NaN	NaN	NaN	0.000000	NaN
50%	NaN	NaN	2.750000	NaN	NaN	NaN	NaN	1.000000	NaN	NaN	0.000000	NaN	NaN	NaN	5.000000	NaN
75%	NaN	NaN	7.250000	NaN	NaN	NaN	NaN	2.625000	NaN	NaN	3.000000	NaN	NaN	NaN	396.000000	NaN
max	NaN	NaN	28.000000	NaN	NaN	NaN	NaN	28.500000	NaN	NaN	67.000000	NaN	NaN	NaN	100000.000000	NaN

[31] Pandas is a Python library for data analysis.

t is also important to know what kind of data/characteristics we will be working with. In this example we are dealing with numeric values (integer numbers, i.e., `int64`, real numbers, i.e., `float64`), and categorical values (classes, i.e., `object`).

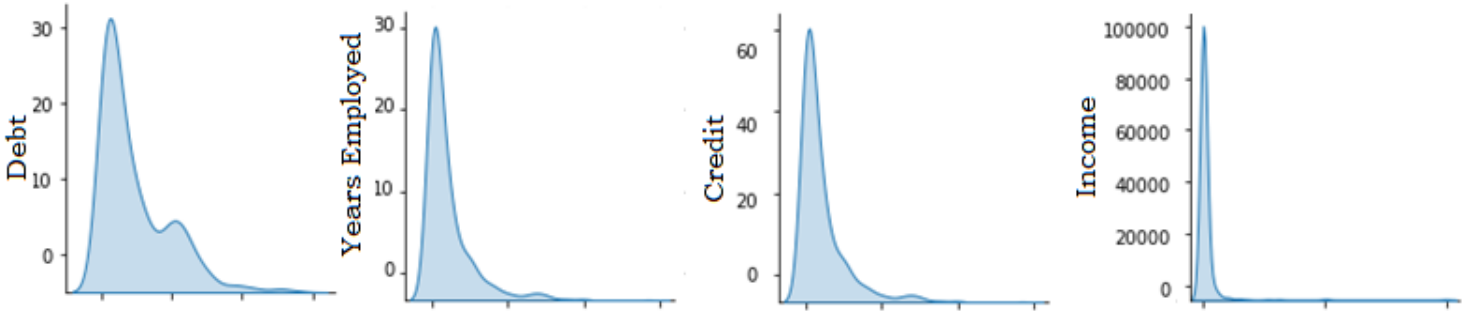
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 689 entries, 0 to 688
Data columns (total 16 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Gender            689 non-null    object
 1   Age               689 non-null    object
 2   Debt              689 non-null    float64
 3   Married           689 non-null    object
 4   Bank Client       689 non-null    object
 5   Education         689 non-null    object
 6   Race              689 non-null    object
 7   Years Employed    689 non-null    float64
 8   Prior Default     689 non-null    object
 9   Employed          689 non-null    object
 10  Credit            689 non-null    int64
 11  Driver's License  689 non-null    object
 12  Citizenship       689 non-null    object
 13  Postal Code       689 non-null    object
 14  Income            689 non-null    int64
 15  Approval Status   689 non-null    object
dtypes: float64(2), int64(2), object(12)
memory usage: 86.2+ KB
```

The dataset used in this example has several missing values (exactly 67) that can hurt the performance of our model. A "best practice" in data science and machine learning is: (1) remove the samples with missing values; or (2) replace the missing values with the mean values (e.g., the Since we are working with a small dataset, we will use practice 2. This is one of the processes we do during preprocessing, in addition to turning all categorical features into numerical features.[32] After this phase, we get a dataset (with no missing values) ready to be used to train a probabilistic classification model.

[32] Logistic regression models will not process categorical variables that are not coded as numbers.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 689 entries, 0 to 688
Data columns (total 16 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Gender            689 non-null    int64
 1   Age               689 non-null    int64
 2   Debt              689 non-null    float64
 3   Married           689 non-null    int64
 4   Bank Client       689 non-null    int64
 5   Education         689 non-null    int64
 6   Race              689 non-null    int64
 7   Years Employed    689 non-null    float64
 8   Prior Default     689 non-null    int64
 9   Employed          689 non-null    int64
 10  Credit            689 non-null    int64
 11  Driver's License  689 non-null    int64
 12  Citizenship       689 non-null    int64
 13  Postal Code       689 non-null    int64
 14  Income            689 non-null    int64
 15  Approval Status   689 non-null    int64
dtypes: float64(2), int64(14)
memory usage: 86.2 KB
```

We can use other tools to explore the data we will be working with. For example, the Seaborn data visualization library can be used to explore the distribution of the data we will be using to train and evaluate our model.



Many of the distributions we have possess “long tails,” i.e., the distribution of values/samples follows a Pareto distribution, i.e., the

volume of samples decreases as the values increase. From this we can interpret that, for example, the vast majority of the samples: (i) has not worked for many years; (ii) has a low credit score; (iii) has a small (or unreported) income.

In other words, our dataset is not “uniform”. It is extremely biased, being a reflection of an unequal environment (e.g., historical biases), and this is a red flag. Our model may come to operate less efficiently when dealing with samples that have not been “seen enough” in its training (i.e., “statistical outliers”). With this in mind, a deeper analysis of the data we will be working with is necessary.

Another tool that we can use to explore the data we are using is the *Facets* library.

Facets[33] s an open-source data visualization tool created by PAIR, designed to aid in the understanding and analysis of datasets used in machine learning. Facets contains two visualization tools:

- *Facets Overview:* Overview provides a quick way to explore the distribution of values between characteristics in a data set (e.g., common/uncommon values, unexpected/absent values, skewness);
- *Facets Dive:* Dive provides an interactive interface to explore the relationship between different characteristics (e.g. how is the distribution of Approval Status versus Gender and Race?) and even individual samples.

Let's see how the distributions analyzed by the Seaborn library are presented by Facets Overview:

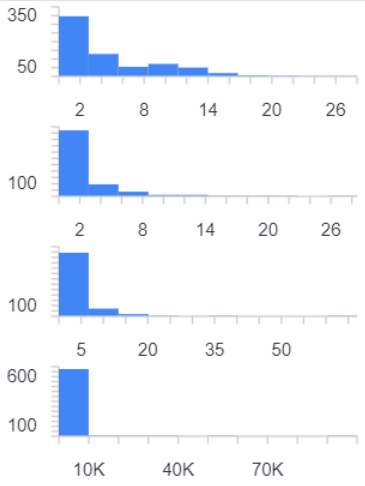
[33] https://Github.com/PAIR-code/facets, https://pair-code.Github.io/facets/.

Sort by
Feature order
Reverse order
Feature search (regex enabled)
Features: int(2) float(2) string(12)

Numeric Features (4)

	count	missing	mean	std dev	zeros	min	median	max
Debt	689	0%	4,77	4,97	2.61%	0	2,75	28
Years Employed	689	0%	2,22	3,35	10.16%	0	1	28,5
Credit	689	0%	2,4	4,86	57.33%	0	0	67
Income	689	0%	1.018,86	5.209,96	42.67%	0	5	100k

Chart to show
Standard
log expand



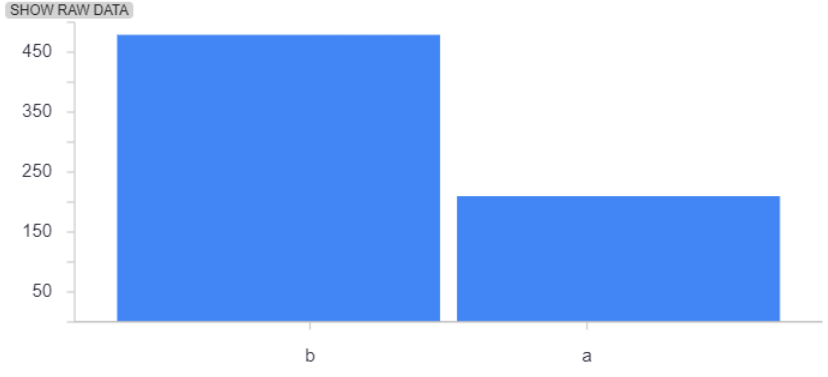
Again, long tails and skewed distributions. It is also curious that 42.67% of the samples have “0” income. If almost half of the samples have a null value, should we use such a characteristic to train our classifier? Another Red Flag.

Categorical Features (12)

	count	missing	unique	top	freq top	avg str len
Gender	689	0%	2	b	479	1

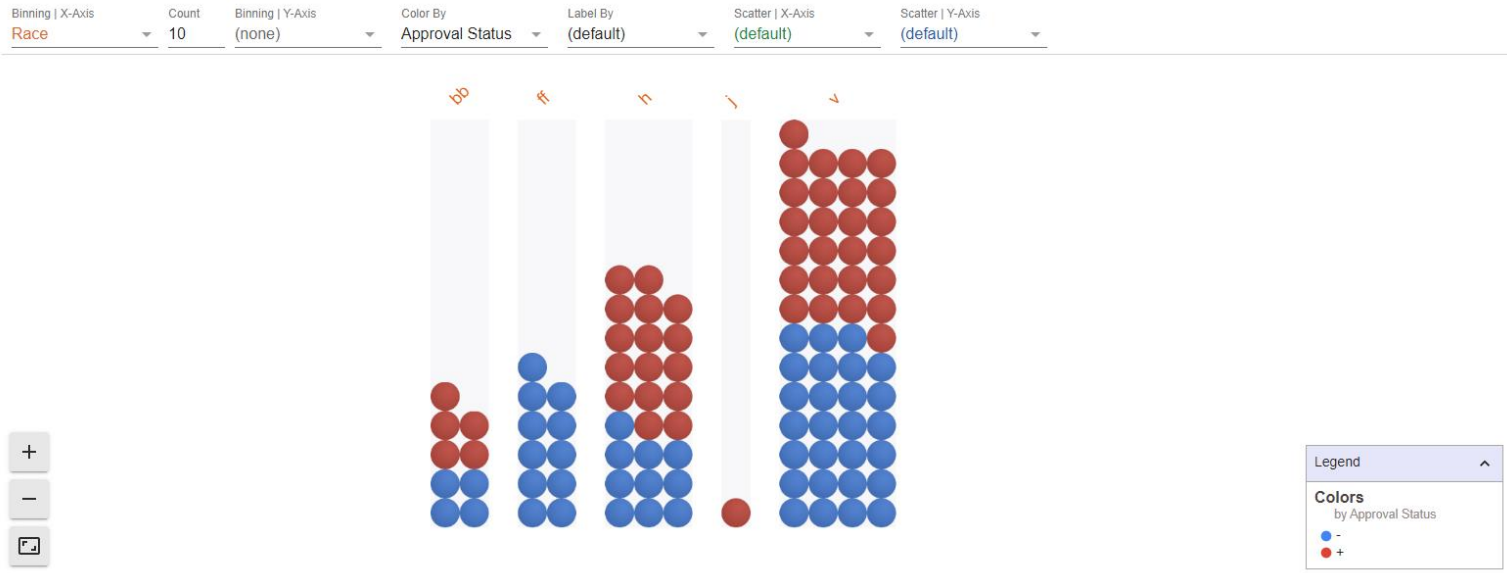
Chart to show
Standard
log expand

SHOW RAW DATA



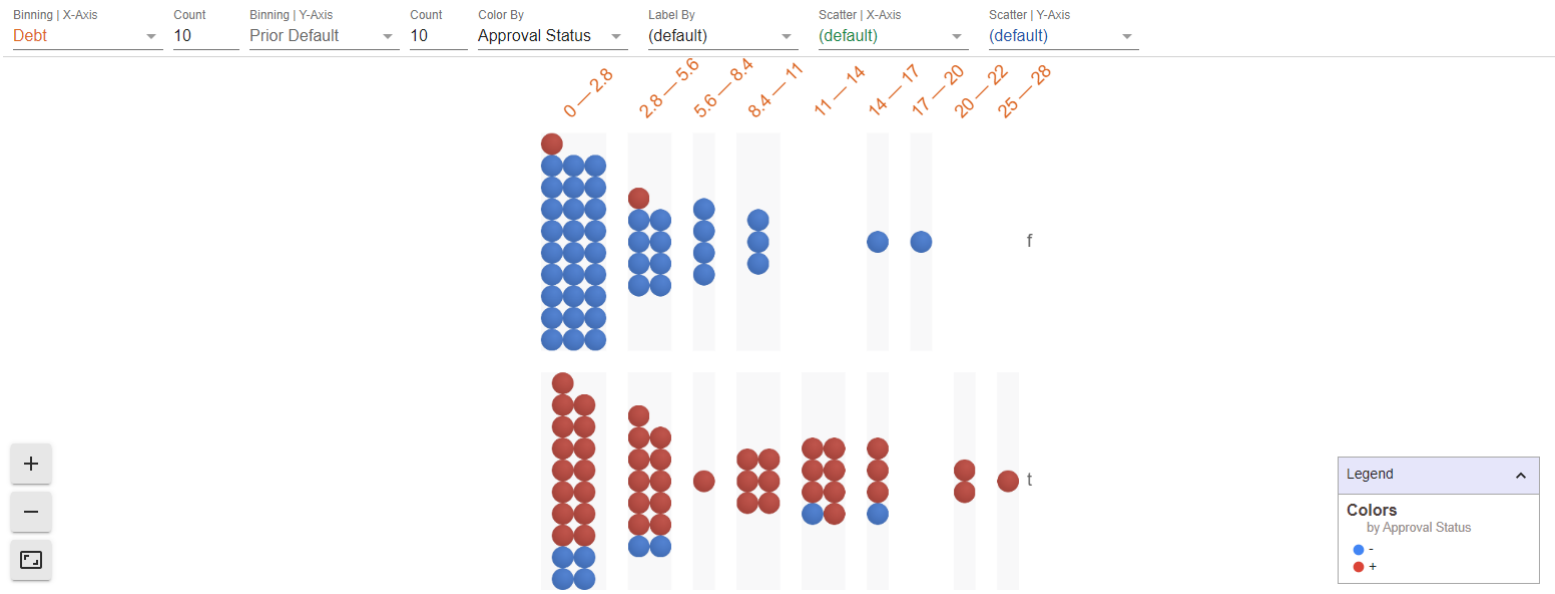
At the same time, more than half of the samples are of a specific gender ("b"). Meanwhile, of the 689 samples, 383 (55.5%) of the credit card applications were denied, and 306 (44.5%) applications were approved. With this data set, we may be creating a model that: (1) does not perform equally (e.g., predictive parity) across genders; and (2) has a higher bias for disapproving applications (How might this affect the bank's customers?).

Now, we dive a little deeper into our dataset with Facets Dive. How does the characteristic "Race" relate to our target (Approval Status)?



Some subgroups of the characteristic "Race" are strongly underrepresented (virtually all when compared to the subgroup "v"). Meanwhile, some subgroups have only negative examples (Not-Approved) while others have only positive examples (Approved).

"Prior Default", i.e., whether the customer has stopped paying bills on other credit cards, should be a determining factor for a credit card application, as should "Debt". How do both of these characteristics relate to Approval Status?



Such characteristics are (apparently) the decisive factors in inferring a sample's Approval Status, since virtually all samples, as their debt increases (0-22), are almost entirely divided between samples that have prior defaults (almost all receive a negative Approval Status) and those with no prior defaults (mostly positive Approval Status).

Data visualization techniques can give us valuable insights into the dataset we are working on, either by detecting possible flaws our model may have, or by deciding which features are best to use in our model. For example, if we adopt a “veil of ignorance” view of fairness, we may choose not to use any sensitive attributes to train our model (e.g., gender, race), since apparently “Prior Default” and “Debt” have a strong correlation with Approval Status.

For simplicity, we will train a generic logistic regression model using scikit-learn, an open-source machine learning library. The data has already been preprocessed (and (re)scaled to small values, i.e., a real number between 0 and 1), and split between a training set (70% of the samples) and a test set (30% of the samples). We will not perform

validation on this example, since it is just an "example". However, real applications need validation steps for tuning the model hyperparameters.

For this example, we will use all 15 features provided by the dataset, as it will be valuable for this study to explore how different features relate to each other, and what coefficients are learned by the model for each feature.

Correlation coefficients measure the linear association between variables/characteristics. We can interpret these values as follows:

- 1: Full positive correlation;
- 0.8: Strong positive correlation;
- 0.6: Moderate positive correlation;
- 0: No correlation at all;
- -0.6: Moderate negative correlation;
- -0.8: Strong negative correlation;
- -1: Total negative correlation.

For example, it is illegal to define the approval status for a credit card application based on the race or gender of the applicant. A positive or negative value of the correlation coefficient of these characteristics with Approval Status would mean unfairness and discrimination by the bank that produced this dataset (something that should not be replicated by any model). Luckily, correlation coefficients can be easily calculated using the NumPy library [34] by the `.corrcoef( )` function.

Correlation Coefficients (Approval Status)	
Gender	0.0300
Age	-0.1300
Debt	-0.2000
Married	0.1900

[34] A library that provides a wide variety of mathematical functions (e.g., multidimensional matrix operations) by high-level commands.

Bank Client	0.1800
Education	-0.1200
Race	0.0003
Years Employed	-0.3200
Prior Default	-0.7100
Employed	-0.4500
Credit	-0.4000
Driver's License	-0.0300
Citizenship	0.1000
Postal Code	0.0900
Income	-0.1700

Fortunately, apparently no sensitive attributes, such as race (0.0003) or gender (0.03), are correlated with Approval Status significantly! In contrast, the characteristic most correlated with Approval Status appears to be Prior Default (-0.7100), something that goes in line with our analysis done using the Facets Dive tool. Apparently, the determining factors for this ranking problem are "Prior Default," "Debt," "Employed" and "Credit." If we determine that such attributes are not sensitive, we could very well train our classifier with only these characteristics, and still get a satisfactory result.

Let's now look at the final result of our model, i.e., its performance with the test portion of the dataset.

Performance (accuracy):		**0.85**
Confusion Matrix	Predicted class (Negative)	Predicted class (Positive)

True Class (Negative)	**94**	**6**
True (Positive) Class	**26**	**102**

We achieved a performance of 85%. Above we also see the confusion matrix from the test we performed of our model. Since we trained our model with more examples of "Reprovals" than "Approvals," we can see that our model has a greater tendency to classify people who should be approved as not-approved (False Negatives = 11%) than to approve people who should be disapproved (False Positives = 0.2%).

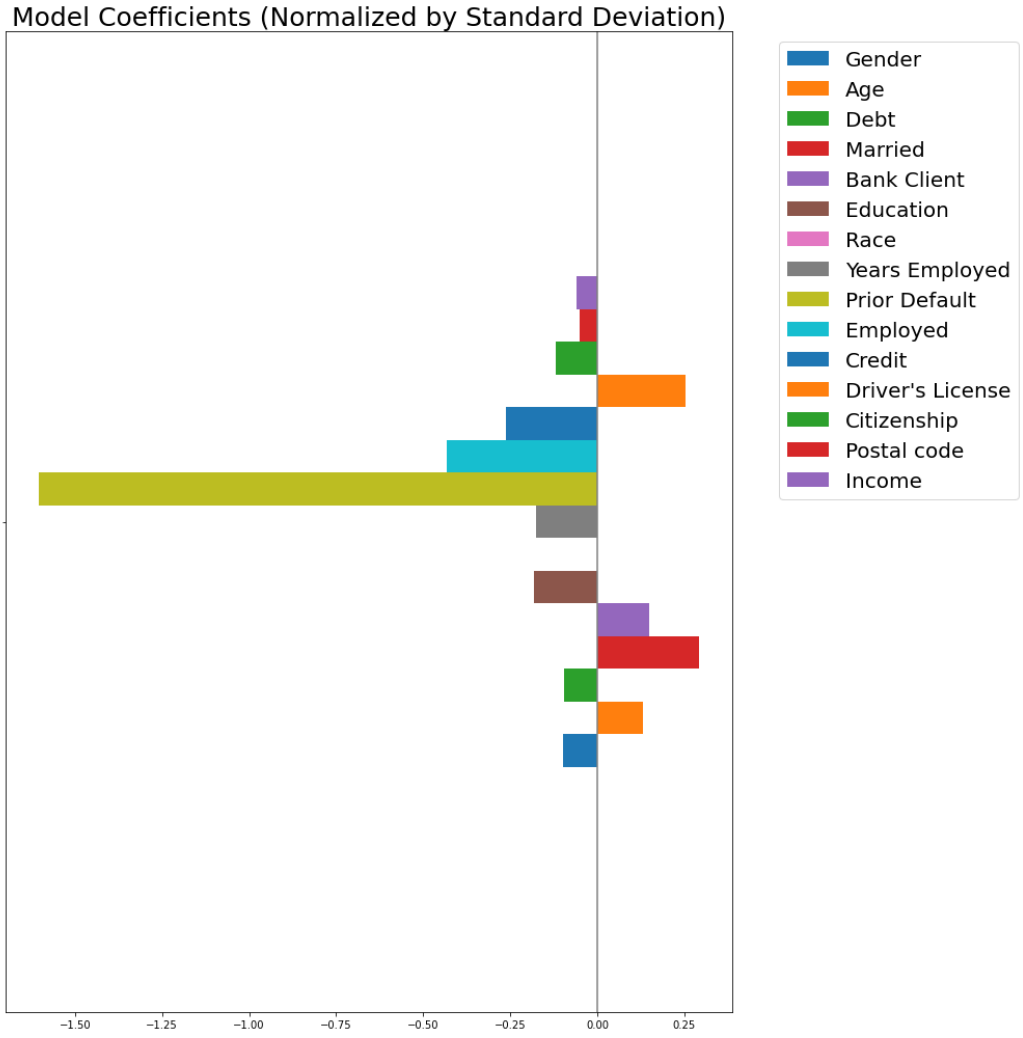
Perhaps, a suggestion to bank managers using this tool is *"Failures should be better investigated/analyzed, you might be losing a good customer."* However, if it is in the bank's interest that False Positives are avoided as much as possible, the trained model has a good ratio between true positives and false positives (i.e., $Precision = 0.94$).

v We will do just one more analysis in this example. We will look at the coefficients learned by our regression model, which basically indicate, as do the correlation coefficients, how much "attention" our model gives to each of the features in a sample.

However, before calculating such coefficients, we need to normalize them. To do this, we will use functions from the Pandas library, `.var( )` and `.std( )`, to calculate the variance and standard deviation of our feature values. [35]

Standard deviation and variance can help us understand other important relationships in our data set. And with standard deviation, we can normalize the coefficients in our model and interpret them correctly (i.e., normalized values are values that "share" a fictitious common scale).

[35] Remember that the variance and standard deviation were calculated with the rescaled/normalized values (delimited between 0 and 1) because it would be meaningless to compare the variance and standard deviation of values measured by different scales (e.g., years versus dollars?).



Again, the main factor for predicting "Approval Status" is "Prior Default". Notice that "Race", with a coefficient of -0.002, is not even visible in the plot above. Armed with all this information, let's now fill in our model card.

Model Card - Credit Card Approval
Model Details
1. Model developed by Nicholas Kluge, researcher at the Pontifical Catholic University of Rio Grande do Sul (PUCRS), in October 2021;

2. This is a Logistic Regression model for binary classification, version 0.1. This model was trained to classify credit card applications as “Not-Approved” or “Approved”;
3. This model was trained only for academic motivations, and it does not follow any kind of fairness/justice constraints. This model is not designed to be implemented in real applications;
4. The dataset used is the Credit Approval Dataset from the UCI Machine Learning Repository. Available at: http://archive.ics.uci.edu/ml/datasets/credit+approval;
5. The code for this model can be found in: https://Github.com/Nkluge-correa/AI-Ethics-exercise;
6. License: MIT License;
7. Contact: nicholas.correa@acad.pucrs.br.

Intended Use

1. The intended use of this model, and the shared code, is to present the developer with tools to explore a dataset and assess possible ethical implications and security flaws of a model trained by machine learning. This model and code are not meant to be used in real applications. However, the tools used can be used for ethical evaluations of models trained by machine learning;
2. This model is designed for the academic audience, developers, and machine learning practitioners interested in learning how to develop “fair” models;
3. As an academic experiment, the only use for this model is to rank credit card applications from samples taken from the Credit Approval Dataset This model should not be used for, e.g., credit score classification, credit score inference, or any other type of task other than its primary intended use.

Factors

1. The characteristics used for the task of rating the Approval Status of a credit card applicant are: “Gender”, “Age”, “Debt”, “Married”, “Bank Client”, “Education”, “Race”, “Years of Employment”, “ Prior Default”, “Employed”, “Credit”, “Driver's License”, “ Citizenship”, “ Postal Code”, “Income”. Attributes like “Gender” and “Race” are considered sensitive attributes;
2. The data used for training does not have an even distribution among the subgroups for each trait. There is a strong bias, for certain types of subgroups, such as genders and specific races.

Metrics

1. The performance metric used was accuracy (total no. of correct classifications per total classifications performed), 85% correct during the test run;
2. The model has a greater tendency to classify people who should pass as failures (False Negatives = 11%) than to approve people who should fail (False Positives = 0.2%);
3. Suggestion: failures should be better investigated/analyzed;

4. Training and testing data were split from the dataset provided by the UCI Machine Learning Repository (i.e., Credit Approval Dataset); 5. This dataset was chosen for its public availability. 6. Samples with missing values (i.e., "?" or "NaN") had such values replaced with the average value of their specific feature.
Ethical Considerations
1. Given the skewed distribution of the training data, the model may behave inefficiently when dealing with poorly seen samples; 2. The model uses sensitive data (i.e., Race and Gender); 3. It's recommended that for real applications, sensitive attributes (e.g., race and gender) and attributes containing "abnormal" values (e.g., income) not be used for classification; 4. According to correlation coefficients, and coefficients learned by the model, sensitive attributes do not interfere with model classification; 5. The attributes most correlated with the applicant's Approval Status are: "Prior Default," "Debt," "Employee," and "Credit."
Details and Recommendations
1. An analysis of the model's performance across different subgroups of each characteristic was not performed. Further analysis may reveal that the model violates fairness criteria, such as predictive parity; 2. The data used for this example do not reflect the social and historical context of a place such as Brazil. They reflect the North American social and historical context. Thus, it is not recommended to use it for application development outside this specific domain.
Quantitative Analysis

	Correlation Coefficients
Gender	0.0300
Age	-0.1300
Debt	-0.2000
Married	0.1900
Bank Client	0.1800
Education	-0.1200
Race	0.0003
Years Employed	-0.3200
Prior Default	-0.7100
Employed	-0.4500
Credit	-0.4000
Driver's License	-0.0300
Citizenship	0.1000
Postal code	0.0900
Income	-0.1700

	Coefficients
Gender	-0.211754
Age	0.476012
Debt	-0.526039
Married	1.753660
Bank Client	0.510739
Education	-0.568626
Race	-0.002892
Years Employed	-0.903885
Prior Default	-3.210696
Employed	-0.879579
Credit	-1.566622
Driver's License	0.508445
Citizenship	-0.413766
Postal code	-0.170551
Income	-1.057520

Performance (accuracy) of the logistic regression model: **0.8596491228070176**

	Predicted Class (Negative)	Predicted Class (Positive)
True Class (Negative)	94	6
True Class (Positive)	26	102

Example 2: Annual Income Forecast

Something we did not do in our last analysis (Example 1) was to evaluate/compare the performance of the trained model between different subgroups:

- *Gender: how does the model performance differ between men and women?*

In this example, we will do exactly this.

We will use the "Adult Census Income Dataset" [36] also provided by the UCI Machine Learning Repository. This dataset is a machine learning "classic" extracted from the US Census Bureau in 1994 by Ronny Kohavi and Barry Becker. The task we will tackle will also be a binary prediction task: *determining whether a person earns more than USD 50,000 per year.*

We will be using virtually all the libraries we used in Example 1 (i.e., Numpy, Pandas, Matplotlib, Seaborn, Facets), with the addition of two new libraries: Tensorflow[37] and Keras.[38] The features contained in this dataset are:

- `"age,"` `"work-class,"` `"fnlwgt"` (the number of individuals the Census Bureau believes the set of observations represents, i.e., the weight of observations), `"education,"` `"education_num"` (an enumeration of the categorical representation of education), `"marital_status,"` `"occupation,"` `"relationship,"`

[36] Lichman, M. (2013). UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/datasets/Census+Income.

[37] An open-source library for machine learning. https://www.tensorflow.org/.

[38] An open-source library, created by François Chollet, for neural network development. https://keras.io/.

"race," "gender," "capital_gain," "capital_loss," "hours_per_week" (hours worked per week), "native_country" (nationality), "income_bracket" (annual income).

We have 14 characteristics and 1 target (i.e., annual income).

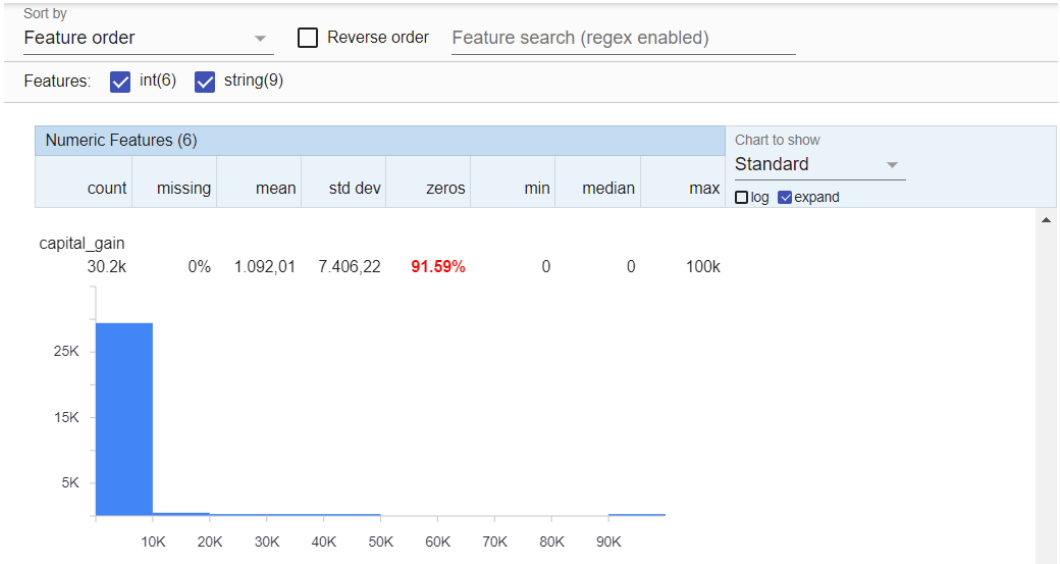
	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relationship	race	gender	capital_gain	capital_loss	hours_per_week	native_country	income_bracket
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
32556	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38	United-States	<=50K
32557	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40	United-States	>50K
32558	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	United-States	<=50K
32559	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	United-States	<=50K
32560	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40	United-States	>50K

With this dataset we have an advantage over the dataset used in the previous example: we have more than 32,000 samples to use. So this time, we will not replace uncommon/absent values (e.g., 'NaN,' '?') with their respective mean values, but we will exclude all samples that have absent values. This leaves exactly 30,163 samples for training (45,224 if we count the samples from the test set). And again, during preprocessing, all (categorical) features will be transformed into numbers and normalized.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             32561 non-null  int64
 1   workclass       30725 non-null  object
 2   fnlwgt          32561 non-null  int64
 3   education       32561 non-null  object
 4   education_num   32561 non-null  int64
 5   marital_status  32561 non-null  object
 6   occupation      30718 non-null  object
 7   relationship    32561 non-null  object
 8   race            32561 non-null  object
 9   gender          32561 non-null  object
 10  capital_gain    32561 non-null  int64
 11  capital_loss    32561 non-null  int64
 12  hours_per_week  32561 non-null  int64
 13  native_country  31978 non-null  object
 14  income_bracket  32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

Before that, let's inspect our dataset directly with Facets, which is (by far) the best data analysis and visualization tool we presented in the previous example. Some questions that can guide our investigation are:

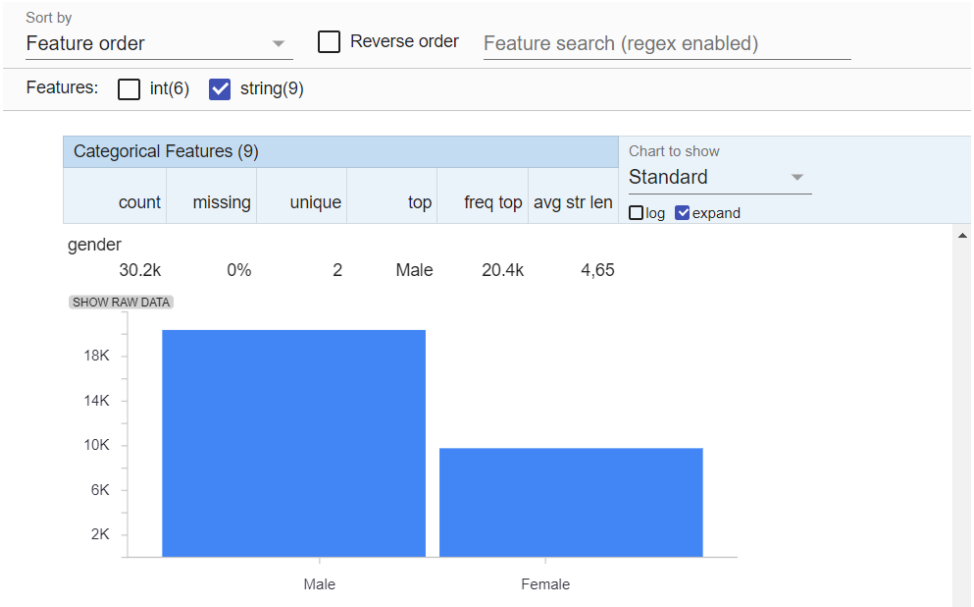
- *Are there missing characteristics that can affect other characteristics?*



Definitely yes. For capital gain/loss/investments, we can see that over 90% of the values are 0. In a world where income distribution is extremely unequal, it should not be a surprise that less than 10% have values other than 0. The vast majority of the population does not invest, gain or lose capital (because they simply do not own it).

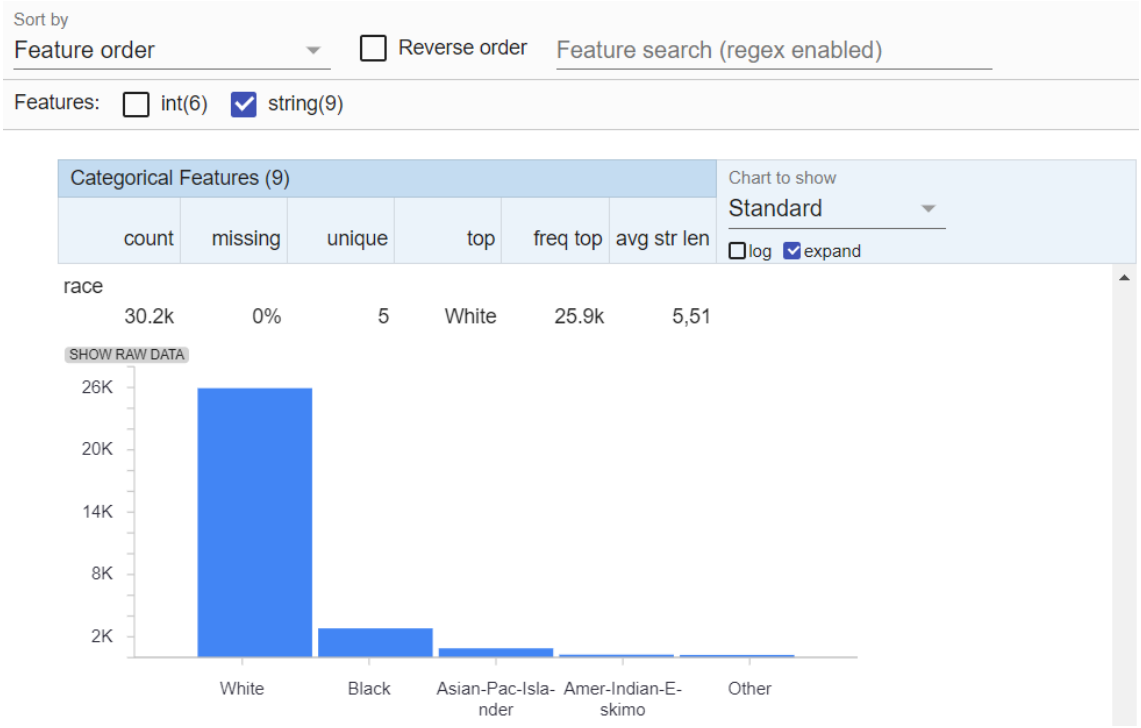
However, it is not at all obvious how to interpret such a result. After all, does "0" mean no gain/loss or unreported gain/loss? Both situations are quite different. In situations like this, it is better not to use such a feature for training our model.

- *Are there signs of bias in the data set?*

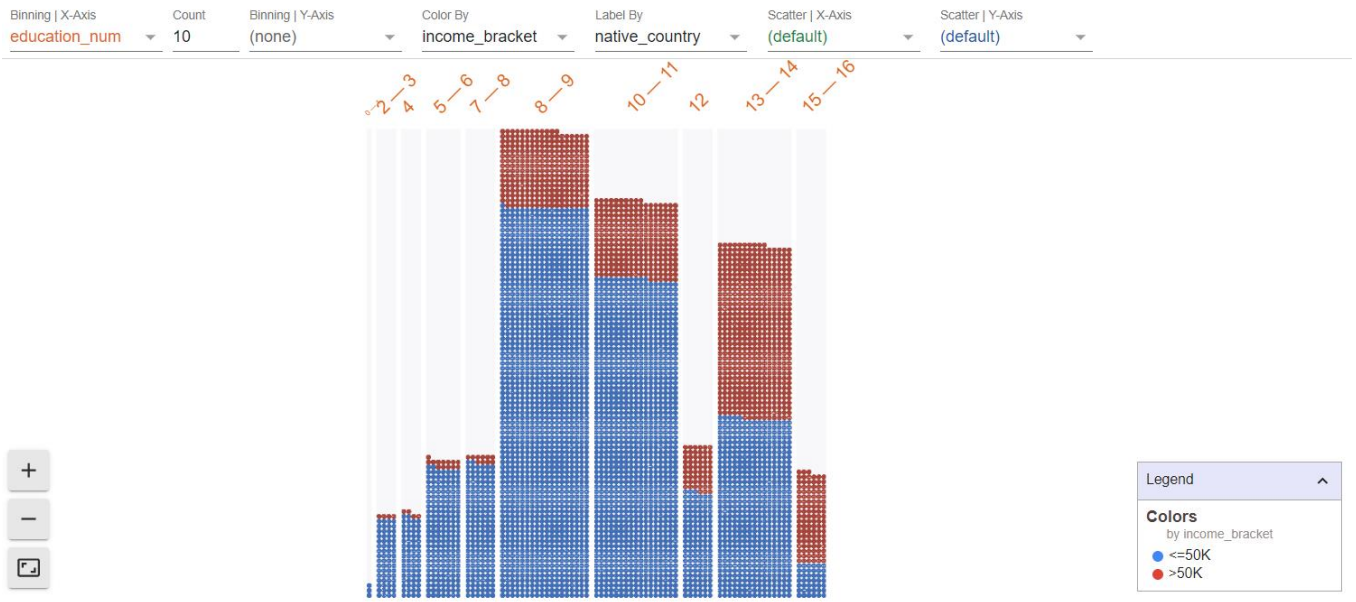


Yes. 67% of the examples represent men. This suggests considerable bias in the data, as we would expect the gender breakdown to be closer to 1:1. In addition to the underrepresentation of the female gender, we see a large racial underrepresentation.

This bias may hurt the performance of our model for a subgroup in which there are few samples/examples.

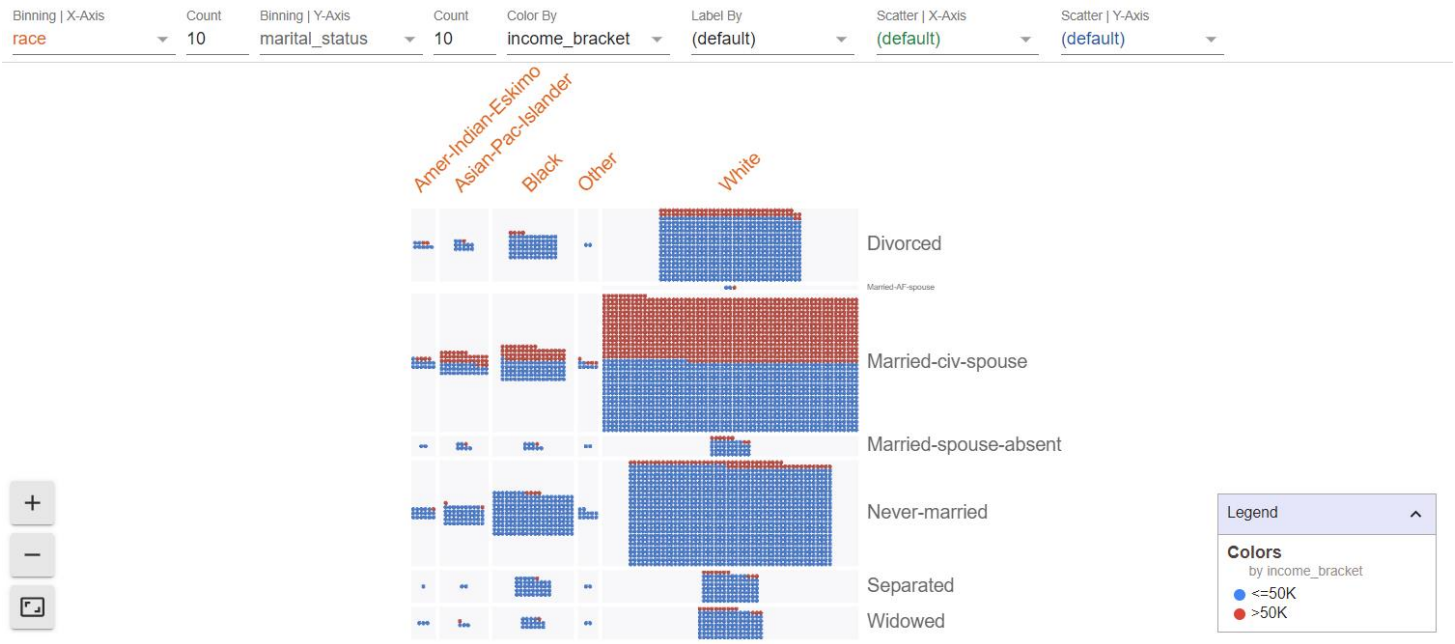


Using Facets Dive, we can look for ways in which characteristics are correlated with each other. Annual income and education level seem to be well correlated, since for the highest levels of education (e.g., Ph.D. and post-doctoral), we see the only class where most samples receive > USD 50,000.



Meanwhile, if we explore Occupation × Gender, we will see that we rarely find women working in the agricultural livestock sector (Could this be a faithful representation of the real world?), while women dominate occupations involving administrative and clerical positions.

There are many other correlations to be investigated, one last one we will show is the intersection of samples between Race × Marital Status × Income.



In a nutshell, if you want to find samples with an annual income of more than USD 50,000, look for married Caucasian people.

For this example, we will only use the following features to train our model:

- `"workclass," "race," "education," "marital_status," "age," "relationship," "native_country," "occupation."`

And we will use the libraries Keras and TensorFlow to create and train a "densely connected feed-forward neural network" with three hidden layers (the tuning parameters of the developed model can be seen in the notebook for this example). We will use 30,163 samples for training and

15,061 samples to test the model (again, since this is just an example, we will skip the validation phase).

In this example, we will use more than one metric to evaluate the performance of our model: accuracy,[39] precision,[40] recall,[41] and AUC.[42] Overall, our model achieves the following performance values:

	Accuracy	**Precision**	**Recall**	**AUC**
Performance	0.8325	0.7074	0.5577	0.8832

Accuracy is the same performance metric that we used in the first example. This is the most "straightforward" and commonly used metric, *"how many times did the classifier get it right?"*. However, accuracy is not always the metric that we should adopt to evaluate a given application.

Precision is generally used as a performance metric for applications where a false positive is a worse problem than a false negative. For example, in spam detection a false positive means blocking a potentially important email. While receiving spam is "tolerable," missing the

[39] The fraction of predictions that a classification model gets right. In binary classification, accuracy has the following definition:

$$acc = \frac{True\ Positives + True\ Negatives}{total\ number\ of\ samples}$$

[40] Precision:

$$pre = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

[41] Recall:

$$rec = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

[42] AUC (Area under the ROC Curve, i.e., a curve of the true positive rate versus the false positive rate at different classification thresholds) is the probability that a classifier is more confident that a randomly chosen positive sample is actually positive than a randomly chosen negative sample is positive.

expected response from that prestigious academic journal is unacceptable.

Recall is the opposite of Precision. Recall measures false negatives against true positives, and in applications such as disease detection, where false negatives must be avoided at all costs, recall is the performance to watch out for.

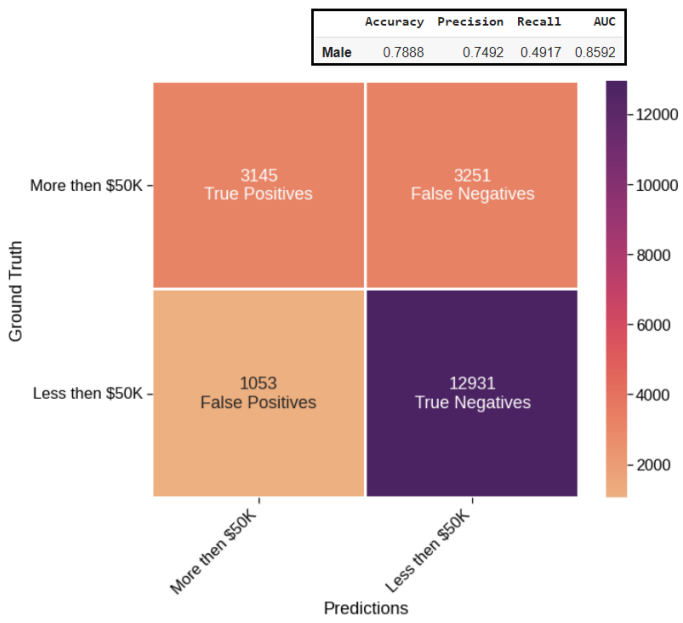
Whereas AUC, which in the case of our model is a metric with the closest value to accuracy, is the probability that, say, our classifier will yell “Wolf!” when there really is a wolf around. That is, to classify a randomly selected sample as its true class.

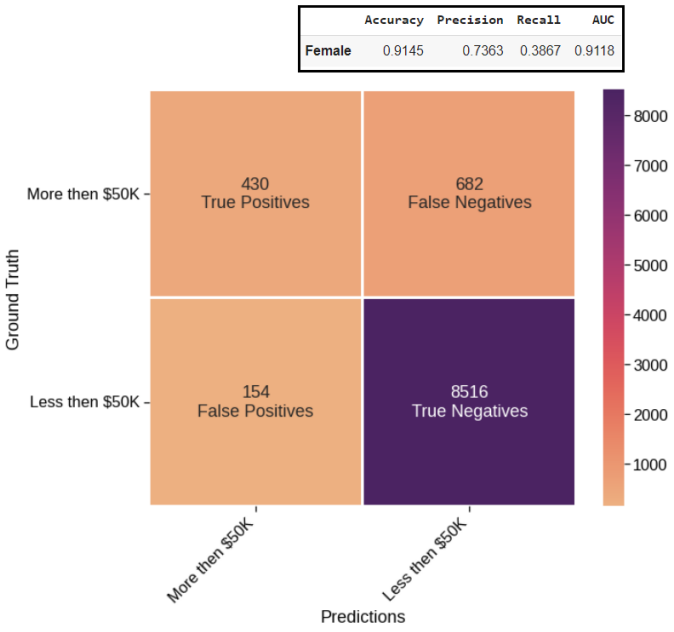
Which metrics should we use to evaluate our model? It depends on the application of this model. Let's say the model will be used to evaluate who (by having an annual income > USD 50,000) should pay more taxes. For that application, a false positive (the individual is classified as receiving > USD 50,000 but actually receives < USD 50,000) seems to be more damaging than a false negative. In other words, for this application, precision seems to be the appropriate performance metric (luckily, the precision of our model is higher than its recall).

While evaluating the overall performance of the model gives us some insight into its quality, it does not give us much insight into the performance of our model for different subgroups. Evaluating a deep neural network is different from evaluating a simple logistic regression model since we cannot inspect the coefficients of this model in an intelligible and simple way (our neural network has more than 35,000 trained parameters).

In this example, we will define that gender, race, and marital status are sensitive attributes. And we will explore some of the differences in

performance between subgroups of these characteristics. If we are going to compare the confusion matrix of the subgroup “Male” vs “Female”:





We will see that in terms of accuracy and AUC, women receive a better rating (accuracy having an almost equal value for both genders). However, since we know that women are disproportionately represented in this dataset, this is a possible sign of overfitting. There is also a

considerable discrepancy in the performance of this model between subgroups of the characteristics race, gender, and marital status.

However, a positive point is that we have, overall, a high precision value in combination with a low recall value. One way to interpret this result is that our classifier is extremely "picky," in the sense that all people classified as "Annual income > USD 50,000" actually have this income. However, the model fails to positively classify several people with income > USD 50,000 because our model is "extremely picky."

If we use this model to define who should pay more (or less) taxes when the model classifies someone as "Annual income > USD 50,000", the model will almost always get it right (the model is accurate). However, many people who also have an income > USD 50,000 will not be "caught" by this classifier.

The summary of the performance of the trained model, across subgroups of the given sensitive attributes, is as follows:

Performance by Gender				
	Accuracy	**Precision**	**Recall**	**AUC**
Male	0.7888	0.7492	0.4917	0.8592
Female	0.9145	0.7363	0.3867	0.9120
Performance by Race				
	Accuracy	**Precision**	**Recall**	**AUC**
Caucasian	0.8227	0.7527	0.4882	0.8812
Black	0.8896	0.7068	0.2568	0.9102
Asian-American	0.7966	0.6774	0.5081	0.8592
Eskimo	0.8951	0.6429	0.2647	0.7831

Others	0.9134	0.5385	0.3333	0.9209
Performance by Marital Status				
	Accuracy	**Precision**	**Recall**	**AUC**
Married (civil spouse)	0.7120	0.7475	0.5541	0.7900
Divorced	0.8949	0.7143	0.0332	0.7959
Married (spouse absent)	0.9189	0.6667	0.0645	0.8214
Never Married	0.9524	1.0000	0.0149	0.8859
Separated	0.9329	0.8000	0.0606	0.8442
Married (military spouse)	0.5238	0.0000	0.0000	0.6955
Widow	0.9033	0.5000	0.0125	0.7569

We cannot attest to statistical parity, predictive parity, or equalized odds for this model. The results show that such a model does not meet these fairness criteria, since, for example, certain subgroups are more susceptible to certain prediction errors than others (especially individuals who belong to certain marital status subgroups, e.g., Married (military spouse).

Such results suggest that we have a model that is overfitted, very much in part by the underrepresentation of several subgroups. Thus, we cannot guarantee that such a model will generalize well, as we do not have enough examples of all subgroups for such a model to "learn."

With all these results in hand, we can now fill out our model card:

Model Card - Annual Income Forecast

Model Details

1. A model developed by Nicholas Kluge, a researcher at the Pontifical Catholic University of Rio Grande do Sul (PUCRS), in October 2021;
2. This is a direct (dense) deep neural network, trained to solve a binary classification task, version 0.1. This model was trained to classify individuals between "Annual income > USD 50,000 " or "Annual income < USD 50,000 ";
3. This model was trained for academic motivations only, and it does not follow any kind of fairness/justice constraints. It is not designed to be implemented in real applications;
4. The dataset used is the Adult Census Income Dataset, made available by the UCI Machine Learning Repository. Available at: http://archive.ics.uci.edu/ml/datasets/Census+Income;
5. The code for this model can be found in: https://Github.com/Nkluge-correa/AI-Ethics-Exercice-2;
6. License: MIT License;
7. Contact: nicholas.correa@acad.pucrs.br.

Intended Use

1. The intended use of this model, and the shared code, is to present the developer with some tools to explore a dataset and evaluate possible ethical implications and security flaws of a model trained by machine learning. This model and code are not meant to be used in real applications. However, the tools used can be used for ethical evaluations of models trained by machine learning;
2. This model was developed for the academic audience, developers, and machine learning practitioners interested in learning how to develop "fair" models;
3. As an academic experiment, the only use for this model is to predict Annual Income from samples taken from the Adult Census Income Dataset. This model should not be used for, e.g., lifetime income prediction, or any other type of task other than its primary intended use.

Factors

1. The characteristics used to train the model are: "work-class," "race," "education," "marital_status," "age," "relationship," "native_country," "occupation." Attributes such as "Gender," "Race," and "Marital Status" were considered as sensitive attributes;
2. The data used for training does not have a uniform distribution among the subgroups for each characteristic. There is a strong bias, for certain types of subgroups, such as gender, marital status, and specific races.

Metrics
1. The performance metrics used were accuracy (83%), precision (70%), recall (55%), and AUC (88%); 2. The model has good accuracy when classifying people who have annual income > USD 50,000 (70%). However, most of the misclassifications made by this model are False Negatives (individuals with annual income > USD 50,000, who are classified as having annual income < USD 50,000; 3. Warning: Model performance varies considerably across subgroups of sensitive attributes (e.g., gender, race, marital status); 4. Training and testing data were acquired directly from the dataset provided by the UCI Machine Learning Repository (i.e., Adult Census Income Dataset); 5. This dataset was chosen for its public availability; 6. Samples with missing values (i.e., "?" or "NaN") were excluded from the dataset.
Ethical Considerations
1. Given the skewed distribution of the training data, the model may behave inefficiently when dealing with poorly viewed samples. Its performance varies considerably between subgroups, failing to reach minimum standards of predictive power for certain subgroups (e.g., Married-military-spouse); 2. It is recommended that for real applications, the dataset be augmented so that there is a better distribution of samples by subgroups of features; 3. According to the performance results and confusion matrices between subgroups, sensitive attributes may interfere with the prediction of this model.
Details and Recommendations
1. The trained model results in a performance that varies across subgroups belonging to sensitive characteristics/attributes. If used for applications that may impact people's lives (e.g., determining who should pay higher taxes), the model may harm underrepresented populations in the Adult Census Income Dataset; 2. The data used for this example does not reflect the social and historical context of a place such as Brazil. They reflect the North American social and historical context. Thus, it is not recommended to use it for application development outside this specific domain.
Quantitative Analysis

Performance by Gender

	Accuracy	Precision	Recall	AUC
Male	0.7888	0.7492	0.4917	0.8592
Female	0.9145	0.7363	0.3867	0.9120

Performance by Race

	Accuracy	Precision	Recall	AUC
Caucasian	0.8227	0.7527	0.4882	0.8812
Black	0.8896	0.7068	0.2568	0.9102
Asian-American	0.7966	0.6774	0.5081	0.8592
Eskimo	0.8951	0.6429	0.2647	0.7831
Others	0.9134	0.5385	0.3333	0.9209

Performance by Marital Status

	Accuracy	Precision	Recall	AUC
Married (civil spouse)	0.7120	0.7475	0.5541	0.7900
Divorced	0.8949	0.7143	0.0332	0.7959
Married (spouse absent)	0.9189	0.6667	0.0645	0.8214
Never Married	0.9524	1.0000	0.0149	0.8859
Separated	0.9329	0.8000	0.0606	0.8442
Married (military spouse)	0.5238	0.0000	0.0000	0.6955
Widow	0.9033	0.5000	0.0125	0.7569

	Accuracy	Precision	Recall	AUC
Performance of the Annual Income Predictor	0.8325	0.7074	0.5577	0.8832

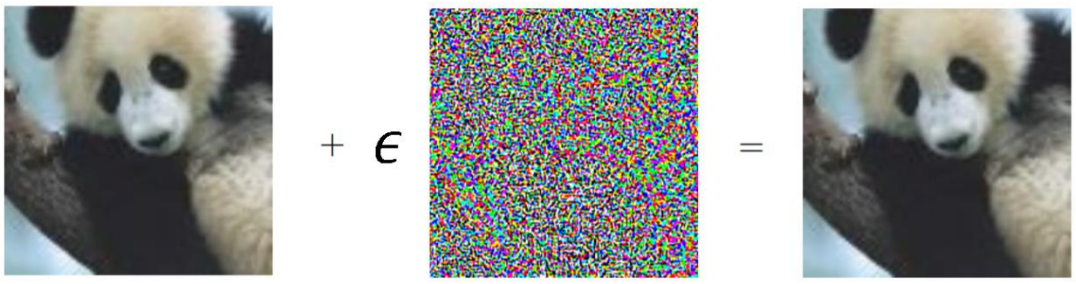
We hope that the examples (as well as the tools) presented in this paper can help developers design and improve their own security analyses, thus

instituting AI Ethics and Security as an integral part of the intelligent systems development process. In the next section, we will present one last methodology to incorporate into a security analysis: *adversarial attacks.*

Adversarial Attacks

Models created by machine learning are curious systems. As much as such systems are capable of performing extremely complex tasks for which we would not know how to "write a solution," their operation and the way such systems "perceive" the environment (i.e., their inputs) allow them to be fooled by what we call "adversarial attacks.

Adversarial attacks, or examples, are inputs/inputs to machine learning models created with the express intent of causing a model to make a mistake (e.g., a misclassification) (Szegedy et al., 2013). These attacks use the fact that machine learning models are (basically) sets of activation functions and parameters optimized by gradient descent. If we have direct (or indirect) access to the parameter values of a model (or the model gradient itself), we can use such information to corrupt input signals by adding almost imperceptible perturbations to make the model produce the output we want.



An adversarial example, created by adding a small perturbation (ϵ) to the image of a "Panda" to make a CNN classify it as a "Gibbon" (Goodfellow et al., 2014, p. 3).

In the example above, Goodfellow et al. (2014) used knowledge of the model's gradient to create an example that (to us) is clearly a panda, but to the model, is a Gibbon with 99.3% confidence. In other words, the authors evaluated how close the "Panda" class is to the "Gibbon" class

within the model's space of representations and "pushed" (i.e., perturbed) such an image to cause the representations/parameters associated with classifying the "Gibbon" class to be strongly (99.3%) activated, causing a misclassification.

With adversarial examples, attackers can exploit potential flaws in models trained by machine learning, something that makes such entities worthy of attention and monitoring. For example, Papernot et al. (2016a) demonstrated how images of traffic signs (e.g., STOP) can be altered to produce misclassifications (e.g., GO), something that could eventually cause traffic accidents involving autonomous cars guided by computer vision. Ahmad et al. (2021) suggest that facial recognition systems used to delimit access to restricted areas could be tricked into allowing unauthorized people to enter (e.g., the attacker can discover a kind of facial makeup/painting that produces a recognition signal with high confidence).

Using the model for credit card approval as an example (Example 1), a simple way to (i) understand how the model works, and (ii) exploit it, is by spoofing signals (i.e., creating adversarial examples). The inputs to the model used in Example 1 are just Rank-1 tensors (i.e., vectors with 15 feature values). So, we can create two Rank-1 tensors (with the appropriate dimension) to test how the model responds. Let's use two extreme examples, i.e., where all values are either 0 or 1:

- `Extreme_case_1 = np.array([[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]]);`
- `Extreme_case_2 = np.array([[1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.]]).`

The "normal" signals/samples have much more varied values (e.g., `sample[10] = ([[0.0, 0.84393064, 0.02982143, 0.5, 0.0, 0.0, 0.88888889, 0.01754386, 0.0, 0.0, 0.0, 1.0, 0.0, 0.49112426,`

0.00228963]])). How does the model created respond to this type of input?

	Approved	**Not Approved**
Extreme_case_1	0.20780003	0.79219997
Extreme_case_2	0.99278847	0.00721153

We now know that samples containing multiple zeros will generate failures (with 79% confidence) and that samples containing multiple ones will generate approvals (with 99% confidence). All we would now need to do is to (subtly) modify the input values to create numerous sample instances that will be classified in any way we wish.

In the battle between attackers and defenders, the defenders are at a disadvantage. Adversarial examples are not necessarily invalid solutions but rather “unexpected solutions” to a complex optimization problem. We use machine learning to find solutions to problems that we do not know how to solve straightforwardly. Since many of the processes that guide the optimization of nonlinear/non-convex problems are not yet fully understood (How can random initialization of a neural network's parameters influence its final performance?) (Frankle & Carbin, 2019), we have no theorems or formal guarantees that allow us to detect/exclude/protect a model against adversarial examples.

Thus, defenders don't have the tools to protect a model against all possible types of attacks because we don't know how to find them systematically and completely. Meanwhile, attackers only need to find “a flaw.” A perturbation that brings them closer to the desired result. And like that, bend the model to their will. Designing defenses against adversarial attacks remains an open problem in AI Safety.

> The study of adversarial examples is exciting because many of the most important problems remain open, both theoretically and in terms of applications. On the

> theoretical side, no one yet knows whether defense against adversarial examples is a theoretically hopeless endeavor (like trying to find a universal machine learning algorithm) or whether an optimal strategy would give the defender some advantage (as in cryptography and differential privacy). On the applied side, no one has yet designed a truly powerful defense algorithm that could withstand a wide variety of adversarial example attack algorithms (Goodfellow & Papernot, 2017).

There are several benchmarks for model robustness evaluation, by which we can perform stress tests and find situations where our models have failed (Hendrycks & Dietterich, 2019; Hendrycks et al., 2021b; Koh et al., 2021). Thus, something a machine learning safety engineer can do is to become the first "attacker" of his own model. In other words, managing adversarial attacks should be one of the essential steps of developing and monitoring a model before and after its deployment.

Much of the current research in adversarial attacks focus on the problem of "l_p adversarial robustness", i.e., situations where attackers seek to induce a model to error but limit the perturbations introduced to the sample within a small constraint ("small perturbations") (Carlini & Wagner, 2017). Attacks can be built on internal model information (e.g., its gradient/parameter values, as was done in the "Panda/Gibbon" example), or just by observing the model's input/output relationship (e.g., as was demonstrated in the credit card approval example) (Tramèr et al., 2018).

There are several strategies for developing adversarial examples, such as brute force search (i.e., massive generation of examples to find adversarial samples), artificial data generation/*data augmentation* (Engstrom et al., 2020; Zhu et al, 2021; Rebuffi et al., 2021), and learning techniques that benefit the detection of samples outside the training distribution and anomalous samples/outliers that are difficult to classify (e.g., *self-supervised learning*) (Hendrycks et al., 2019).

For those interested in learning more about techniques for building adversarial examples, CleverHans[43] is a software library that provides standardized reference implementations to help developers create models that are more robust to adversarial samples. Using CleverHans, developers can create their own adversarial datasets in a standardized way and train their models to handle such samples robustly. Developers can even create their own evaluation/training benchmarks against adversarial samples (Papernot et al., 2016b).

Ian Goodfellow and Nicolas Papernot (creators of the CleverHans library) maintain a blog [44] about safety and privacy in machine learning. There you can find commented examples, along with open-source scripts, teaching developers how to perform security analysis.

[43] https://Github.com/cleverhans-lab/cleverhans.

[44] http://www.cleverhans.io/.

Closing Remarks

Importantly, to date, there is little evidence that the use of any of the tools/methods mentioned in this work are effective in optimizing the ethical design of algorithmic systems. As such, it is still necessary for studies aimed at implementing these techniques to demonstrate the results of their methodologies, either by assisting disadvantaged social groups or avoiding possible side effects of poorly designed AI systems.

The main goal of this guide is to provide developers of AI systems with tools and methods to apply during the life cycle of these types of systems. It is only through experimentation that we will know which tools work, which work better, and which should be improved.

We hope that we have helped all those interested in bridging the gap between theory and practice of safe and ethical AI development to broaden their knowledge.

References

Agüera y Arcas, B., Todorov, A., & Mitchell, M. (2018). Do algorithms reveal sexual orientation or just expose our stereotypes? *Medium*. https://link.medium.com/GO7FJgFgM1.

Ahmad, Z., Khan, A. S., Shiang, C. W., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1), e4150. doi: 10.1002/ett.4150.

AI Robotics Ethics Society (AIRES) at PUCRS. (2021). An Open Letter to the Global South: Bring the "rest" in. AI Robotics Ethics Society.

AlgorithmWatch. (2020). AI Ethics Guidelines Global Inventory. Algorithm Watch. https://inventory.algorithmwatch.org/.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete problems in AI safety. ArXiv. https://arxiv.org/abs/1606.06565.

Ananny, M., & Crawford, K. (2018). Seeing without knowing: Limitations of the transparency ideal and its application to algorithmic accountability. *New Media & Society*, 20(3), 973–989. doi:10.1177/1461444816676645.

Badia, A., Bilal, P., Kapturowski, S., Sprechmann, P., Vitvitskyi, A., Guo, D., & Blundell, C. (2020). Agent57: Outperforming the Atari Human Benchmark. DeepMind. https://arxiv.org/pdf/2003.13350.pdf.

Balch, O. (2020). AI and me: friendship chatbots are on the rise, but is there a gendered design flaw? *The Guardian*. https://www.theguardian.com/careers/2020/may/07/ai-and-me-friendship-chatbots-are-on-the-rise-but-is-there-a-gendered-design-flaw.

Baum, S. (2017). A Survey of Artificial General Intelligence Projects for Ethics, Risk, and Policy. Global Catastrophic Risk Institute, Working Paper, 1-17. http://dx.doi.org/10.2139/ssrn.3070741.

Bender, E. M., & Friedman, B. (2018). Data Statements for NLP: Toward Mitigating System Bias and Enabling Better Science. *Transactions of the Association for Computational Linguistics,* 6, 587-604. doi: 10.1162/tacl_a_00041.

Boddington, P. (2017). *Towards a code of ethics for artificial intelligence*. Springer International Publishing. doi: 10.1007/978-3-319-60648-4.

Bonilla-Silva, E. (2013). *Racism without Racists: Color-Blind Racism and the Persistence of Racial Inequality in the United States* (4th edition). Rowman & Littlefield Publishers.

Brown, T., Mann, B., Ryder, N., et al. (2020). Language Models are Few-Shot Learners. ArXiv. https://arxiv.org/pdf/2005.14165.pdf.

Calo, R. (2017). Artifcial intelligence policy: a primer and roadmap. *SSRN Journal*, 399–435. doi:10.2139/ssrn.3015350.

Calvo R. A., Peters D., Vold K., & Ryan R. M. (2020) *Supporting Human Autonomy in AI Systems: A Framework for Ethical Enquiry*. *In Ethics of Digital Well-Being, Philosophical Studies Series*, vol 140, Burr C., & Floridi L. (eds.). Springer, Cham. doi: 10.1007/978-3-030-50585-1_2.

Carlini, N., & Wagner, D. (2017). Towards evaluating the robustness of neural networks. In the *2017 IEEE Symposium on Security and Privacy*. https://arxiv.org/abs/1608.04644.

Carrillo, R. M. (2020). Artificial intelligence: From ethics to law. *Telecommunications Policy*, 44(6), 101937. doi: 10.1016/j.telpol.2020.101937.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H., Kaplan, J., Edwards, H., Burda, Y., Joseph, N. et al. (2021). Evaluating Large Language Models Trained on Code. OpenAI. https://arxiv.org/abs/2107.03374.

Chouldechova, A. (2016). Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. *Big Data*, 5(2), 153-163. doi:10.1089/big.2016.0047.

Churchland, P. S., & Sejnowski, T. (1992). *The computational brain*. USA, Cambridge: MIT Press.

Collins, E. (2018). Punishing Risk. *Geo. L. J*, 57. https://ssrn.com/abstract=3171053

Corbett-Davies, S., Pierson, E., Feller, A., Goel, S., & Huq, A. (2017). Algorithmic Decision Making and the Cost of Fairness. In *the Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 797–806. doi:10.1145/3097983.3098095.

Corrêa, N. K., & De Oliveira, N. (2021). Good AI for the Present of Humanity Democratizing AI Governance. *AI Ethics Journal*, 2(2)-2. doi: 10.47289/AIEJ20210716-2.

Critch, A., & Krueger, D. (2020). AI Research Considerations for Human Existential Safety (ARCHES). ArXiv. https://arxiv.org/abs/2006.04948.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zeme, R. (2012). Fairness Through Awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference,* 214–226. doi: 10.1145/2090236.2090255.

Ekstrand, M.D., Joshaghani, R., & Mehrpouyan, H. (2018). Privacy for all: Ensuring fair and equitable privacy protections. *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, 1–13.

Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Steinhardt, J., & Madry, A. (2020). Identifying Statistical Bias in Dataset Replication. In 2020 *International Conference on Machine Learning.* https://arxiv.org/abs/2005.09619.

Everitt, T., Kumar, R., Krakovna, V., Legg, S. (2019). Modeling AGI Safety Frameworks with Causal Influence Diagrams. DeepMind. https://arxiv.org/abs/1906.08663.

Fazelpour, S., & Lipton, Z. C. (2020). Algorithmic fairness from a non-ideal perspective. In the *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 57–63. doi: 10.1145/3375627.3375828.

Fitzgerald, M., Boddy, A., & Baum, S. B. (2020). 2020 Survey of Artificial General Intelligence Projects for Ethics, Risk, and Policy. Global Catastrophic Risk Institute Technical Report 20-1.

Fjeld, J., Achten, N., Hilligoss, H., Nagy, A., & Srikumar, M. (2020). Principled Artificial Intelligence. Mapping Consensus in Ethical and Rights-Based Approaches to Principles for Ai. In Berkman Klein Center Research Publication 2020, p. 1–39.

Floridi, L., & Taddeo, M. (2016). What is data ethics? Philosophical Transactions of the Royal Society A: Mathematical, *Physical and Engineering Sciences*, 374(2083), 20160360. doi: 10.1098/rsta.2016.0360.

Floridi, L., Cowls, J., Beltrametti, M., et al. (2018). AI4People—An Ethical Framework for a Good AI Society: Opportunities, Risks, Principles, and Recommendations. *Minds & Machines*, 28, 689–707. doi:10.1007/s11023-018-9482-5.

Frankle, J., & Carbin, M. (2019). The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In the *International Conference on Learning Representations (2019).* https://openreview.net/pdf?id=rJl-b3RcF7.

Fryer, R., Loury, G., & Yuret, T. (2008). An Economic Analysis of Color-Blind Affirmative Action. *Journal of Law, Economics, and Organization*, 24(2), 319–355.

Gajane, P., & Pechenizkiy, M. (2018). On Formalizing Fairness in Prediction with Machine Learning. Department of Computer Science, Montanuniversitat Leoben, Austria, and the Department of Computer Science, TU Eindhoven, the Netherlands. ArXiv. https://arxiv.org/abs/1710.03184.

Galhotra, S., Brun, Y., & Meliou, A. (2017). Fairness Testing: Testing Software for Discrimination. In *the Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, 498–510. doi:10.1145/3106237.3106277.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumé III, H., & Crawford, K. (2018). Datasheets for Datasets. ArXiv. https://arxiv.org/abs/1803.09010.

Goldsmith, J., & Burton, E. (2017). Why teaching ethics to AI practitioners is important. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 4863–4840. https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14271/13992.

Goodfellow, I. J., Shlens, J., Szegedy, C. (2014). Explaining and Harnessing Adversarial Examples. In the *2015 International Conference on Learning Representations*. https://arxiv.org/abs/1412.6572.

Goodfellow, I., & Papernot, N. (2017). Is attacking machine learning easier than defending it? Cleverhans-blog. www.cleverhans.io/security/privacy/ml/2017/02/15/why-attacking-machine-learning-is-easier-than-defending-it.html.

Green, B. (2019). "Good" isn't good enough. In *NeurIPS workshop on AI for social good*. https://www.benzevgreen.com/wp-content/uploads/2019/11/19-ai4sg.pdf.

Greene, J., Rossi, F., Tasioulas, J., Venable, K. B., & Williams, B. (2016). Embedding Ethical Principles in Collective Decision Support Systems. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16). https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12457.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv. CSUR*, 51(5), 93:1–93:42. doi: 10.1145/3236009.

Hagendorff, T. (2020). The Ethics of AI Ethics: An Evaluation of Guidelines. *Minds and Machines*, 30, 99–120. doi:10.1007/s11023-020-09526-7.

Hardt, M., Price, E., & Srebro, N. (2016). Equality of opportunity in supervised learning. In *the Proceedings of the 2016 Advances in neural information processing systems*, 29, 3315–3323.

Hendrycks, D., & Dietterich, T. (2019). Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In the *2019 Proceedings of the International Conference on Learning Representations*. https://arxiv.org/abs/1903.12261.

Hendrycks, D., Mazeika, M., Kadavath, S., & Song, D. (2019). Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty. In the 2019 *Conference on Neural Information Processing Syste*ms. https://arxiv.org/abs/1906.12340.

Hendrycks, D., Carlini, N., Schulman, J., Steinhardt, J. (2021a). Unsolved Problems in ML Safety. ArXiv. https://arxiv.org/abs/2109.13916#.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., & Gilmer, J. (2021b). The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. In the *2021 International Conference on Computer Vision.* https://arxiv.org/abs/2006.16241.

Hirose, I. (2014). *Egalitarianism* (1st edition). UK, London: Routledge.

Hofstede, G. H., Hofstede, G. J., & Minkov, M. (2010). *Cultures and Organizations: Software of the Mind* (3rd edition). New York, NY: McGraw-Hill.

Holland, S., Hosny, A., Newman, S., Joseph, J., & Chmielinski, K. (2018). The Dataset Nutrition Label: A Framework To Drive Higher Data Quality Standards. ArXiv. https://arxiv.org/abs/1805.03677#.

Hubinger, E., van Merwijk, C., Mikulik, V., Skalse, J., & Garrabrant, S. (2019). Risks from Learned Optimization in Advanced Machine Learning Systems. Machine Intelligence Research Institute. https://arxiv.org/abs/1906.01820.

Hutter, M. (2005). Universal artificial intelligence: Sequential decisions based on algorithmic probability. *Springer-Verlag Berlin Heidelberg*. doi:10.1007/b138233.

Jobin, A., Ienca, M., & Vayena, E. (2019). The global landscape of AI ethics guidelines. *Nat Mach Intell*, 1, 389–399. doi:10.1038/s42256-019-0088-2.

Jurić, M., Šandić, A., & Brcic, M. (2020). AI safety: state of the field through quantitative lens. *43rd International Convention on Information and Communication Technology, Electronics and Microelectronics* (MIPRO). https://arxiv.org/ftp/arxiv/papers/2002/2002.05671.pdf.

Kärkkäinen, K., & Joo, J. (2019). FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age. University of California, Los Angeles. ArXiv. https://arxiv.org/abs/1908.04913.

Kenton, Z., Everitt, T., Weidinger, L., Gabriel, I., Mikulik, V., Irving, G. (2021). Alignment of Language Agents. DeepMind. https://arxiv.org/abs/2103.14659.

Kilbertus, N., Rojas-Carulla, M., Parascandolo, G., Hardt, M., Janzing, D., & Schölkopf, B. (2017). Avoiding Discrimination Through Causal Reasoning. In *the Proceedings of the 31st International Conference on Neural Information Processing Systems*, 656–666. doi:10.5555/3294771.3294834.

Kleinberg, J., Mullainathan, S., & Raghavan, M. (2016). Inherent trade-offs in the fair determination of risk scores. Cornell University and Harvard University. ArXiv. https://arxiv.org/abs/1609.05807.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Beery, S., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., & Liang, P. (2021). WILDS: A Benchmark of in-the-Wild Distribution Shifts. In the *2021 International Conference on Machine Learning.* https://arxiv.org/abs/2012.07421.

Krafft, T. D., & Zweig, K. A. (2019). Transparency and traceability of algorithm-based decision-making processes | A regulatory proposal. Verbraucherzentrale Bundesverband (Federal Association of Consumer Organizations). https://www.vzbv.de/sites/default/files/downloads/2019/05/02/19-01-22_zweig_krafft_transparenz_adm-neu.pdf.

Krafft. T. B., Hauer, M., Fetic, L., Kaminski, A., Puntschuh, M., Otto, P., Hubig. C., Fleischer, T., Grünke, P., Hillerbrand, R., Husted. C., & Hallensleben, S. (2020). From Principles to Practice - An interdisciplinary framework to operationalise AI ethics. AI Ethics Impact Group (VDE Association for Electrical, Electronic & Information Technologies/Bertelsmann Stiftung). https://www.ai-ethics-impact.org/en.

Krishnan, M. (2019). Against Interpretability: A Critical Examination of the Interpretability Problem in Machine Learning. *Philosophy and Technology*, 33(1). doi: 10.1007/s13347-019-00372-9.

Leike, J., Martic, M., Krakovna, V., Ortega, P. A., Everitt, T., Lefrancq, A., Orseau, L., Legg, S. (2017). AI Safety Gridworlds. DeepMind. https://arxiv.org/abs/1711.09883.

Lohr, S. (2018). Facial Recognition Is Accurate, if You're a White Guy. *The New York Times*. https://www.nytimes.com/2018/02/09/technology/facialrecognition-race-artificial-intelligence.html.

Luengo-Oroz, M. (2019). Solidarity should be a core ethical principle of AI. *Nat Mach Intell*, 1(494). doi:10.1038/s42256-019-0115-3.

Luong, B. T., Ruggieri, S., & Turini, F. (2011). k-NN As an Implementation of Situation Testing for Discrimination Discovery and Prevention. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 502–510.

Maxmen, A. (2018). Self-driving car dilemmas reveal that moral choices are not universal. *Nature*, 562 (7728), 469–470. doi:10.1038/d41586-018-07135-0.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2019). A survey on bias and fairness in machine learning. *ACM Computing Surveys*, 54(6), 1-35. doi:10.1145/3457607.

Mikulik, V., Delétang, G., McGrath, T., Genewein, T., Martic, M., Legg, S., Ortega, P. A. (2020). Meta-trained agents implement Bayes-optimal agents. DeepMind. https://arxiv.org/abs/2010.11223.

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P. Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru, T. (2019). Model Cards for Model Reporting. In the *Proceedings of the Conference on Fairness, Accountability, and Transparency* (January, 2019), 220–229. doi:10.1145/3287560.3287596.

Mittelstadt, B., Russell, C., & Wachter, S. (2019). Explaining explanations in AI. In *FAT* '19: Proceedings of the Conference on Fairness, Accountability, and Transparency,* 279–288. doi:10.1145/3287560.3287574.

Mnhi, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning. DeepMind. https://arxiv.org/abs/1312.5602.

Morley, J., Floridi, L., Kinsey, L., & Elhalal, A. (2019). From what to how: An initial review of publicly available AI ethics tools, methods and research to translate principles into practices. *Science and Engineering Ethics*, 26(4), 2141–2168. doi: 10.1007/s11948-019-00165-5.

Morley, J., Elhalal, A., Garcia, F., Kinsey, l., Mökander, J., & Floridi, L. (2021). Ethics as a service: a pragmatic operationalisation of AI Ethics. *Minds and Machines*, 31,239–256. doi:10.1007/s11023-021-09563-w.

Newell, A. (1990). *Unified theories of cognition*. USA, Cambridge: Harvard University Press.

Nunes, P. (2019). EXCLUSIVO: levantamento revela que 90,5% dos presos por monitoramento facial no Brasil são negros. *The Intercept Brasil*. https://theintercept.com/2019/11/21/presos-monitoramento-facial-brasil-negros/.

Papernot, N., Faghri, F., Carlini, N., Goodfellow, I., Feinman, R., Kurakin, A., Xie, C., Sharma, Y., Brown, T., Roy, A., Matyasko, A., Behzadan, V., Hambardzumyan, K., Zhang, Z., Juang, Y., Li, Z., Sheatsley, R., Garg, A., Uesato, J., Gierke, W., Dong, Y., Berthelot, D., Hendricks, P., Rauber, J., Long, R., & McDaniel, P. (2016a). Technical Report on the CleverHans v2.1.0 Adversarial Examples Library. ArXiv. https://arxiv.org/abs/1610.00768.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, B. Z., & Swami, A. (2016b). Practical Black-Box Attacks against Machine Learning. In the *Proceedings of the 2017 ACM Asia Conference on Computer and Communications Security, Abu Dhabi, UAE.* https://arxiv.org/abs/1602.02697.

Pearl, J. (1995). *Causation, Action, and Counterfactuals*. In *Computational Learning and Probabilistic Reasoning*, A. Gammerman (ed.), USA, New York: John Wiley and Sons, 235–255.

Rahimi, A [Preserve Knowledge]. (2018, March 7). NIPS 2017 Test of Time Award "Machine learning has become alchemy." | Ali Rahimi, Google [Video]. Youtube. https://www.youtube.com/watch?v=x7psGHgatGM.

Rawls, J. (1999). A Theory of Justice. UK, Oxford: Oxford University Press.

Rebuffi, S., Gowal, S., Calian, D. A., Stimberg, F., Wiles, O., & Mann, T. A. (2021). Fixing Data Augmentation to Improve Adversarial Robustness. In the *2021 Conference on Neural Information Processing Syste*ms. https://arxiv.org/abs/2103.01946.

Rességuier, A., & Rodrigues, R. (2020). AI ethics should not remain toothless! A call to bring back the teeth of ethics. *Big Data & Society*, 1-5. doi:10.1177/2053951720942541.

Russell, S., Dewey, D., & Tegmark, M. (2015). An Open Letter: Research Priorities for Robust and Beneficial Artificial Intelligence. Open Letter. Signed by 8,600 people. https://futureoflife.org/data/documents/research_priorities.pdf.

Ruster, L. (2021). Dignity & Artificial Intelligence: Exploring the role of dignity in government AI ethics instruments. Centre for Public Impact. https://www.centreforpublicimpact.org/partnering-for-learning/cultivating-a-dignity-ecosystem-in-government-ai-ethics-instruments.

Saravanakumar, K. K. (2021). The Impossibility Theorem of Machine Fairness -- A Causal Perspective. Columbia University. ArXiv. https://arxiv.org/abs/2007.06024.

Sen, A. (1990). Justice: Means versus Freedoms. *Philosophy and Public Affairs*, 19.

Silver, D., Huang, A., Maddison, C., et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature,* 529, 484–489. doi:10.1038/nature16961.

Silver, D., Singh, S., Precup, D., & Sutton, R. S. (2021). Reward is enough. *Artificial Intelligence*, 299(103535). doi:10.1016/j.artint.2021.103535.

Soares, N. (2016). Value Learning Problem. In *Ethics for Artificial Intelligence Workshop, 25th International Joint Conference on Artificial Intelligence* (IJCAI-2016), USA, New York 9–15. https://intelligence.org/files/ValueLearningProblem.pdf.

Soares, N., Fallenstein, B., Yudkowsky, E., & Armstrong, S. (2015). Corrigibility. In *Artificial Intelligence and Ethics*, T. Walsh (ed.), AAAI Technical Report WS-15-02. Palo Alto, CA: AAAI Press.

Suresh, H., & Guttag, J. (2021). A Framework for Understanding Potential Sources of Harm throughout the Machine Learning Life Cycle. *MIT Case Studies in Social and Ethical Responsibilities of Computing*. doi:10.21428/2c646de5.c16a07bb.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R. (2013). Intriguing properties of neural networks. In the *2014 International Conference on Learning Representations*. Intriguing properties of neural networks. https://arxiv.org/abs/1312.6199.

Tramèr, F., Kurakin, A., Papernot, N., Boneh, D., & Mcdaniel, P. (2018). Ensemble Adversarial Training: Attacks and Defenses. In the *2018 International Conference on Machine Learning.* https://arxiv.org/abs/1705.07204.

Verma, S., & Rubin, J. (2018). Fairness definitions explained. In *the Proceedings of the 2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*, 1–7.

Wang, Y., & Kosinski, M. (2017). Deep neural networks are more accurate than humans at detecting sexual orientation from facial images. doi:10.1037/pspa0000098.

Wilkinson, M. D. et al. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data*, 3(160018). doi:10.1038/sdata.2016.18.

Wolf, M., Miller, K., & Grodzinsky, F. (2017). Why we should have seen that coming: comments on microsoft's tay experiment, and wider implications. *ACM SIGCAS Computers and Society*, 47(3), 54–64.

Ye, W., Liu, S., Kurutach, T., Abbeel, P., & Gao, I. (2021). Mastering Atari Games with Limited Data. In the *35th Conference on Neural Information Processing Systems (NeurIPS 2021).* https://arxiv.org/abs/2111.00210.

Yuan, X., He, P., Zhu, Q., & Li, X. (2019). Adversarial Examples: Attacks and Defenses for Deep Learning. *IEEE Trans. Neural Netw. Learn. Syst.*, 30(9), 2805–2824. doi:10.1109/TNNLS.2018.2886017.

Zhu, Y., Ma, J., Sun, J., Chen, Z., Jiang, R., & Li, Z. (2021). Towards Understanding the Generative Capability of Adversarially Robust Classifiers. In the *2021 International Conference on Computer Vision.* https://arxiv.org/abs/2108.09093.