center for responsible ai

APR 2023

ALGORITHMIC TRANSPARENCY PLAYBOOK

A stakeholder-first approach to creating transparency for your organization's algorithms.

Andrew Bell
Julia Stoyanovich, Ph.D.
Oded Nov, Ph.D.

Contents

Welcome to the Playbook

In recent years, there has been a rapid proliferation of algorithmic tools into the public and private sectors to improve processes and increase efficiency. **Algorithmic decision-making systems (ADS)** are systems that use algorithms for making decisions in a specific context, such as finance, employment, healthcare, and education.

While these algorithmic tools have the ability to greatly improve society, they also have the potential to cause great harm. As a case study, Amazon once built and implemented an automated resume screening and hiring system—only to later find out that the **system was biased against hiring women** (7). In another instance, algorithmic systems threatened global economic stability by causing the 2010 Flash Crash, wherein **erroneous decisions** made by complex algorithmic trading systems caused the Dow Jones to lose $1 trillion in value in 36 minutes (11). Both of these issues were caused, in part, by a lack of **transparency** into the underlying algorithmic system.

In this playbook, we address risks like those just described by providing instructions, best-practices, and recommendations on algorithmic transparency. We do this by thinking primarily about the stakeholders of algorithmic decision-making, that is, the individuals or groups - both internal and external to an organization - that are impacted by an algorithmic system.

What will this playbook teach you?

By the time you finish reading the playbook, you will be able to answer the following questions:

- What is algorithmic transparency and how is it defined?
- Who are the stakeholders of algorithmic transparency?
- What are the goals of algorithmic transparency?
- What are the existing methods for algorithmic transparency?
- What are the best practices for designing and implementing algorithmic transparency into existing and future algorithmic systems?
- How can algorithmic transparency be maintained into the future?
- How can I help influence a culture shift in my organization towards adopting algorithmic transparency?

You will also learn about several useful case studies in algorithmic transparency.

Who should read this playbook?

This playbook is intended for a range of audiences:

- *C-suite executives and managers.* In organizations using algorithmic decision-making systems, it's critical that leadership understands how important algorithmic transparency is, both from a business and ethical perspective. Executives and managers should understand what algorithmic transparency is, why it is useful for different stakeholders, and how it can be implemented within their organization.
- *Policymakers and regulators.* In recent years, governments around the world have begun to regulate algorithmic systems, often requiring some degree of transparency. Unfortunately, all existing and emerging legislation on algorithmic transparency to date share a common weakness: it has focused on what to do (or what not to do), but has left the brunt of the work to data scientists to figure out how (9). By reading this playbook, policymakers can better

understand concrete ideas about algorithmic transparency, and ergo better understand how it can be regulated.

- *Data scientists and data/software engineers.* This playbook is useful for learning state-of-the-art and industry-standard practices for algorithmic transparency. By reading this playbook you will learn new methods for transparency and have an understanding of how to implement transparency in the most effective way possible in your organization.

All About Transparency

How do we define "algorithmic transparency?"

Algorithmic transparency is the principle that information about decisions made by algorithms should be visible to those who use, regulate, and are affected by the systems that employ those algorithms. Algorithmic transparency should enable the monitoring, checking, criticism, or intervention by interested parties. Generally, algorithmic transparency *is about the human understanding of algorithms*.

Note that there is no single there is no single definition for algorithmic transparency. The definition we provide here is an amalgamation from Policy Briefs on algorithmic transparency and academic work. The definition is intentionally very broad, and can represent multiple meanings.

For example, in one situation, algorithmic transparency may mean that **people are able to anticipate what decision an algorithmic system would make**, and in another context it could mean **knowing the list of factors that are taken into account by the algorithmic system.**

As an analogy, consider that there are many ways to understand how a television set works. You can understand aspects like, - What it does (ex. displays a picture) - How to work it (ex. using the remote) - How it works to the extent that you could fix it if it breaks or reconstruct it from the ground up

There Is No One Term To Rule them All

It's important to note that data scientists, researchers, managers, and even policymakers use a variety of definitions (and often interchange them) to speak about transparency. For example, researchers and data scientists most commonly use the term **explainability** to mean transparency, but other terms include *interpretability, understandability, intelligibility, comprehensibility, accountability, traceability*, and *legibility* (16).

What are some examples of algorithmic transparency?

Algorithms have completely profileated the public and private sector, and ultimately our lives. Algorithms determine which movies are recommended to us on Netflix, what pages are at seen at the top of your Google search result, and even makes recommendations on what word we should type next in a text message. Many of these algorithms are not transparent, but here are 3 examples of algorithmic systems that *do* have transparency in the wild:

- *Hiring.* It is increasingly common for companies to use *algorithmic hiring tools* that automatically process and screen resumes that they receive from online job applicants. These tools are meant to "read" through resumes and select the best candidates for interviews. In New York City, it is now law that companies **must** inform job applicants that their resume is being processed by an algorithm rather than a human.
- *Finance.* many banks uses algorithms to determine if an applicant should be accepted or denied for a loan. By law, if an algorithm recommends to reject an applicant, that algorithm must provide *reason codes* that are given to the applicant, such as "Too many recently opened bank accounts with balances."
- *Education.* Many schools across the US have begun using algorithms to identify students that are risk of failing out of school. These systems are called *Early Warning Systems*, and provide a score to school administrators and teachers for how likely a student is to fail. To support transparency, the algorithms behind the Early Warning Systems are made extremely simple, and normally involve adding up aspects like "how many times has the student been absent?",

"how many classes have they failed in the past", and "how many times has the student been sent to detention?"

Why is algorithmic transparency critically important?

- **Transparency can play a critical role in avoiding the significant risks and harms associated with algorithmic decision systems.** These risks include performance risks like algorithmic errors, security risks, control risks like rogue outcomes and unintended consequences, economic risks, ethical risks, and societal risks like unfair outcomes for underprivileged or marginalized communities. For companies, this often means avoiding significant *costs and fines*.
- **Transparency may improve your algorithmic systems.** Transparency can increase trust among all the stakeholders of an system, which may lead to better outcomes for that system.
- **Implementing transparency will soon likely be law.** Governments around the world have begun to draft and enact legislation regulating the transparency of algorithmic systems, including two major proposals in both the EU and US.

Who is transparency for?

Algorithmic transparency is important for many people both internally to and externally of an organization. There are five main groups of people impacted by algorithms:

- *Practitioners.* The technical practitioners that are developing, implementing, and maintaining algorithmic systems. They include *data scientists, engineers, programmers, developers, and analysts.*
- *Managers.* Those that oversee projects using algorithmic tools. They exist at many many different levels in an organization, and include *project managers, business developers, procurement manager, and executives.*
- *Affected persons.* The people who are impacted by the algorithm. For example, if an algorithm is being used to assess job applicants, the job applicants are the affected persons.
- *Humans-in-the-loop.* The individuals who are responsible for *using the algorithm.* Humans-in-the-loop may also be called algorithm managers or users.
- *Regulators.* Persons who oversee the legal compliance of algorithms, and may include also include *auditors, compliance officers, and policymakers.*

Is algorithmic transparency easy to do?

The short answer is yes. For any algorithm, there are easy first steps that can be taken to move from an algorithm being to *completely opaque* to *at least slightly tranparent.*

The Easiset Fist-Step for Transparency

A first step in algorithmic transparency can be be *as simple as telling people that you are using an algorithm.* Many organizations either take this fact for granted, or deliberately keep this fact hidden from the public. Do not underestimate the value of simply disclosing the use of an algorithmic system, and telling people what attributes, factors, or inputs the algorithm uses.

The long answer is that, while it can be easy to create basic transparency features, it takes more effort to create robust and lasting transparency. One should be thoughtful about *how, why and to what end* transparency is implemented — all of which will be discussed extensively in this course. Importantly, we believe that the organizational and societal benefits of algorithmic transparency greatly justify the efforts of implementing it correctly.

I've heard of algorithms called "black-boxes." Can they be made transparent?

Yes. In recent years, researchers and practitioners have made significant developments in "opening up" complex, opaque systems called *black-boxes*. There are many powerful, free, and easy-to-use tools that can be used to gain insight into how underlying black-box algorithms work. For example, a tool you may often hear about is "SHapley Additive exPlanations **(SHAP)**." It's not important to understand SHAP on a technical level, but know that it can be used to create easily understandable explanations of how complex, black-box systems make decisions at both the **local** and **global** level (15).

> The Two Levels of Transparency
>
> **Local transparency** refers to understanding how an algorithmic system makes a decision about a single case or instance, and **global transparency** refers to understanding how the system works overall. For example, for an algorithmic system that predicts whether or not an applicant is accepted for a loan, global transparency describes how the entire system works, and local transparency describes the system's prediction for a single loan applicant.

How is algorithmic transparency different from fairness or bias?

Algorithmic transparency is often confused with ideas like **fairnessand bias** — but importantly, they are not the same thing.

Algorithmic transparency is about the visibility of information about a system. In contrast, *algorithmic fairness* is concerned with addressing societal biases that are present in an algorithmic system. While fairness and transparency can be related (for example, visibility into an algorithmic system may reveal its biases), they are actually independent. Ensuring transparency does not ensure fairness, and vice versa. Instead, they are connected in-so-far as it is infinitely easier to detect fairness issues in algorithms that are transparent.

Barriers to Algorithmic Transparency

It's important to ask, "why don't companies make algorithmic transparency a priority?"

There are several key reasons why organizations avoid or neglect algorithmic transparency, which we list in the section. Importantly, we also offer a rebuttal for each reason, and why they are inadequate justification for not pursuing transparency. These rebuttals can be useful to help you become a *transparency infleuncer* within your organization, and create meaningful change towards having a more transparent, ethical approach to algorithms.

Claim: Transparency Means More Costs

This is not necessarily true.

While it is true that are some costs to implementing algorithmic transparency, **these costs are often grossly overstated — especially when compared to the potential costs of *not* implementing transparency.** Later in this course, we will detail the exact process and types of conversations one needs to have to implement transparency.

Notably, what is often *understated* is the costs that are saved through algorithmic transparency. Transparency can be used to avoid performance risks like algorithmic errors, security risks, control risks like rogue outcomes and unintended consequences, economic risks, ethical risks, and societal risks like unfair outcomes for underprivileged or marginalized communities. All of these risks can have costly consequences, like poor algorithms leading to worse business decisions or public relations risks, which can be greatly mitigated by using transparency.

> Case Study: The Cost of Not Being Transparent
>
> In 2019, Meta (then Facebook) received a record-breaking FTC fine of $5 billion for violations related to privacy, accountability and transparency. Despite the fine being substantial, the true penalty is that Meta is now required to hire compliance officers that actively participate and oversee business decisions at *all levels of the company* — a cost that is even more impactful than the $5 billion fine. This case study illustrates a key point: **ignoring transparency may save costs in the short-run, but leaves an organization vulnerable to catastrophic risks in the future.**

Claim: Transparency Means Less Accurate Algorithms

This is not true.

Many managers are concerned that implementing transparency means reducing the sophistication of their algorithmic systems, thereby decreasing their efficiency and accuracy. For example, in the context of algorithmic hiring, some managers erroneously believe that making a resuming screening tool more transparent means it will perform worse.

However, recent research has challenged this idea. First, as described previously, there are a number of transparency tools that can be used to "open up" even the most complex, sophisticated black-box systems. Second, there is a growing number of case studies showing that under many conditions, simpler, more transparent algorithmic systems can perform the same (or even better than) complex systems. **Overall, implementing transparency does not necessarily result in sacrificing efficiency–it's not that simple!**

Claim: Transparency Means Open-Sourcing Algorithms

This is not true.

Algorithmic transparency is **not** the same as "open-sourcing" technologies. While providing the source code for an algorithm does offer some transparency into how it works, it is not necessary.

In fact, in many cases, open-sourcing is an insufficient or misguided attempt at transparency for two reasons: first, the source code is not useful for laymen or any non-technical stakeholders of the algorithm in helping them understand how it works. Second, the source code for an algorithm is only one component of a much larger technical ecosystem. Without the data that is used by the algorithm or the technical infrastructure that supports it, the source code may be completely useless.

Note that in some situations open-sourcing *can* be a component of transparency, but it is by no means required.

Claim: Transparency Means Losing Intellectual Property (IP)

Losing IP is not a guaranteed part of transparency.

Protecting IP is often a major concern for small startups or companies whose main competitive advantage is due to the IP of their algorithms. It would be untruthful to claim that transparency is not, at least in some ways, in conflict with protecting IP. However, we forward the claim that *it is possible* to implement elements of transparency without significantly jeopardizing the privacy of an organization's IP. In some ways, there is a balancing act to perform between taking advantage of the benefits of algorithmic transparency, while protecting IP. We offer the following ideas for transparency when IP protection is also a consideration:

- **Creating transparency for different pieces of elements or your algorithm, where the sum of those pieces are insufficient to full reconstruct your IP.** It is not uncommon that an algorithm is made up of multiple layers of decision making, or uses tens (or hundreds) of attributes, factors, and inputs to make decisions. Perhaps it is possible to implement transparency surroundings specific layers of the algorithm, or focusing on just 3-5 factors (or broad categories of factors).
- **Using Differential Privacy.** Differential privacy is a system for publicly sharing information about a dataset by describing the patterns of groups within the dataset, while withholding the true information contained in a dataset.

Claim: Transparency Means Sacrificing Privacy

This is rarely true.

Privacy in the context of algorithmic systems is generally concerned with protecting sensitive data that the organization has collected. We want to make it clear that **it is a complete myth** that one has to sacrifice the privacy of their data to offer transparency. It is *never* necessary to expose sensitive or proprietary data to ensure algorithmic transparency. In fact, this is exactly the objective of *Differential Privacy.*

Differential Privacy

Differential privacy is a large anc complex topic, but a primer can be found here.

Claim: Transparency Means Strategic Manipulation

Unfortunately, even without algorithmic transparency, strategic manipulation of algorithms by users is widespread. For example, in the hiring space, some candidates add invisible keywords in white text to the bottom of their resumes to trick algorithms into scoring their application higher. Another more prevalent example is Search Engine Optimization (SEO), which is the process of trying to "game" your webpage to the top of the search results.

In light of this, transparency may actually be an end-run around strategic manipulation. **There is substantial data showing that algorithmic transparency increases the trust of users (5)**. It's not unreasonable to believe that if users trust a system, a good faith dialogue can be opened up about preventing strategic manipulation and the abuse of those systems.

When Algorithmic Tools Are Procured

Many organizations, especially in governmental or intergovernmental organizations, choose to procure their algorithmic tools instead of building them in-house. This poses a unique challenge, because because it may be beyond the agency or control of your organization to implement transparency for these tools. **The extent to which your organization values having agency over algorithmic transparency should influence whether or not it procures algorithmic tools.**

We have also drafted this list of probing questions that can be asked to organizations providing algorithmic tools that will help open the conversation around transparency:

- What are your values around algorithmic transparency?
- What considerations to transparency have you implemented in the tool being considered?
- If we require additional transparency considerations, are you able to implement them?
- What transparency is available to the organization *selling* the tool, that is not available to the *procuring* organization (and ultimately those that are impacted by the tool)?
- How much transparency can we pass down stream to those impacted by the algorithmic tool?

Case Study: Model Cards for AI Transparency

Salesforce is a Fortune 500 company that is well known for selling software tools to business and non-profits. Notably, in 2020, they began producing model cards for their algorithmic tools. **Model cards** provide high-level details about an underlying algorithm like when and where it was created, its intended primary and secondary uses, what factors the algorithm considers, and against which metrics the performance of the tool was evaluated.

The Algorithmic Transparency Checklist

Our framework for algorithmic transparency is made up of 4 steps, which are be described in detail in the playbook: Inventory, Plan Design, Implement, and Maintain.

Using this checklist is as easy as going through each item in order. If you need further clarification on any step or sub-step, see the detailed instructions found in this playbook.

STEP 1: Inventory your algorithmic systems	
	1A: Create a list of all algorithmic decision-making systems in your organization. Note that the definition of an algorithmic decision-making system is broad, and includes any system that uses an algorithm for making decisions.
STEP 2: Plan & Design transparency for your algorithmic systems	
	2A: Create a list of all the relevant stakeholders for each algorithmic system. The stakeholders for algorithmic systems include technical practitioners, managers, affected persons, humans-in-the-loop, and compliance officers. Stakeholders exist both inside and outside your organization.
	2B: Create a list of the potential goals of each stakeholder. The goals of transparency are ensuring validity, building trust, assisting in learning and support, supporting recourse, and ensuring fairness and privacy. Note that stakeholders may have different or overlapping goals for transparency. Ideally, you should engage stakeholders and have them talk about their goals.
	2C: Design transparency features for your systems given stakeholders' goals. Consider transparency features like transparency labels, data visualizations, intrinsic transparency mechanisms of algorithmic systems, and attribute importance, and attribute influence. As a baseline, you should strongly consider implementing transparency labels, attribute importance, and attribute influence for algorithmic systems impacting people and their lives.
	2D: Speak with your technical team to review your design ideas. There is an information asymmetry between what is technical feasible and transparency and what is ideal for stakeholders, making it important to include the technical team in transparency discussions early on.
STEP 3: Implement transparency for your algorithmic systems.	
	3A: Implement transparency features. This is a technical and design process that may include creating data visualizations, dashboards, or algorithmic factsheets.
	3B: Test and review your transparency implementation with stakeholders. To avoid the pitfalls of poorly implemented transparency features, you should consult with stakeholders to make sure that transparency features are implemented properly.
STEP 4: Maintain algorithmic transparency	
	4A: Create a process for continually maintaining algorithmic transparency. This includes implementing transparency from the outset when creating new algorithmic system, and monitoring transparency features to make sure they don't degrade over time.

STEP 1: Inventory your algorithmic systems

The first step in implementing algorithmic transparency is to understand everywhere that algorithmic decision-making systems are being used within your organization.

Step 1A: Create a list of all the algorithmic decision-making systems in your organization

The definition of an **algorithmic system** is quite broad, and includes any system that relies on the analysis of data to derive algorithms for making decisions. Other names for these tools are *automated decision systems, artificial intelligence (AI) systems, machine learning (ML) systems, or data science systems.* In general, algorithmic tools receive a data input, process that input through an algorithm, and then produce an output in the form of a decision.

> **Interchangeable Terms for Algorithmic Decision-Making Systems**
>
> From here forward, we will refer to algorithmic decision-making systems in shorthand as either *algorithmic decision systems, algorithmic systems*, or just simply *algorithms*. In all of these cases, we are referring to the decision-making system as a whole.

Examples of algorithmic systems include hiring tools, predictive algorithms, and smart metering systems. Predictive algorithms have a wide range of applications including predicting student performance, chance of hospital re-admission, risk of diseases, whether not a loan applicant should be accepted or rejected, or the likelihood a user will click on an advertisement.

When inventorying your organization for algorithms, keep these two important points in mind:

- **Algorithms are not always big, complex systems.** Sometimes algorithmic systems can be deceptively simple — and its important not to overlook them! For example, many schools use *Early Warning Systems* that only look at three factors (a students' grades, attendance record, and the number of detentions they received) which are tallied together, sometimes on paper and without the help of a computer. Despite its simplicity, this is still an algorithm!
- **Algorithms in procured tools.** Many organizations, especially in governmental or intergovernmental organizations, choose to procure their algorithmic tools instead of building them in-house. Don't forget to include these in your inventory! Even if they are not internal to your organization, there are still actions towards transparency for these type of tools.

> **DELIVERABLE**
>
> An inventory list of all the algorithms in your organization.

STEP 2: Plan design transparency for your algorithms

Once you have identified all the algorithmic decision-making systems within your organization, it is time to begin planning and designing how transparency will be used in those algorithms. In the case of procured tools, these steps will likely involve engaging with the organization that owns the algorithm.

This second step is broken up into 4 sub-steps and is based on our **stakeholder-first approach** to developing transparency that begins with thinking about algorithmic stakeholders first, and ends with creating transparency features for your algorithms that meet stakeholder needs.

A **transparency feature** is any artifact accompanying an algorithm that helps increase its ability to be understood by a human user. Some examples include a dashboard, graph, report, paper hand-out, or even an informational pop-up window on a web page.

Step 2A: Consider all the relevant stakeholders of each algorithm.

In the introduction to this playbook, we discussed several important stakeholders of algorithmic decision-making systems. These included **managers** within an organization, the **humans-in-the-loop** who actually use the systems, and the **affected persons** who are impacted by the outcome of the stakeholder. Notably, stakeholders may be both internal and external to your organization (1; 17; 18).

Humans-in-the-loop, or the people who are actually responsible for using the algorithmic tool. These are often distinct from those developing the algorithm, and some examples include an underwriter using an algorithm to determine if loan applicants should be accepted or rejected, or hospital staff using an algorithm to predict the risk a patient will develop a disease.

Importantly, not every stakeholder has the same needs when it comes to algorithmic transparency, and so those implementing transparency should be thoughtful about each type of stakeholder. There are 5 categories of stakeholders that you should consider (note that not every algorithm will have all 5 stakeholders):

Stakeholder	**Definition**
Practitioners	The technical practitioners that are developing, implementing, and maintaining algorithmic systems. They include *data scientists, engineers, programmers, developers,* and *analysts.*
Managers	The individuals at many different levels in an organization that oversee algorithmic decision-making tools. They include *project managers, business developers,* and *executives.*
Affected persons	The people who are impacted by the algorithm. For example, if an algorithm is being used to assess job applicants, the job applicants are the affected persons.
Humans-in-the-loop	The individuals who are responsible for using the algorithm. Humans-in-the-loop may also be called *algorithm managers* or *users.*
Compliance officers	Persons who oversee the legal compliance of algorithms, and may include *auditors* and *policymakers.*

For each algorithm you found in **Step 1A**, you should consider **each of the stakeholder categories above.** Furthermore, you may want to prioritize your list of stakeholders and weigh their needs differently. For example, it may be more meaningful to meet the transparency needs of affected persons over managers or compliance officers.

DELIVERABLE

A list of stakeholders for each algorithmic decision-making system in your organization.

Step 2B: Create a list of the potential goals of each stakeholder.

After you have determined the stakeholders of each system in your organization, you should consider their goals for transparency. Remember that transparency goals always start with a stakeholder since transparency is always ultimately intended for a human audience.

Broadly, the goals of transparency are ensuring validity, building trust, assisting in learning and support, supporting recourse, and ensuring fairness and privacy (17; 16). These goals are below:

Goal	Definition	Example
Validity	Making sure the system is constructed correctly, debugging a system	The programmers, engineers, and managers may use transparency to ensure the system is valid and correct
Trust	Knowing "how often the system is right"	A policymaker or auditor may use transparency to gain trust in the ADS
Learning and support	Increasing general understanding about how an algorithm reaches a decision	A doctor may use transparency to better understand an algorithms predicted diagnosis of a patient
Recourse	Allowing affected persons to take action against a decision	An individual may use transparency about an algorithm to appeal a loan rejection
Fairness	Ensuring that an algorithm is not making decision biased against a minority group	An auditor may use transparency to make sure that an algorithm is not biased
Privacy	Ensuring that an algorithm respects the data privacy of individuals	An auditor may use transparency to make sure that an algorithm is not violating data privacy laws

One critical goal for transparency is the idea of **recourse** (sometimes called redress), which is the ability of a person affected by the outcome of an algorithm to see why that decision was made and what they can do to change that outcome. For example, if an algorithm is used to determine whether or not an individual is accepted or rejected for a loan, that individual should be able to see why that decision was made so they can take actions to change the decision in the future (ex. improve credit score). Notably, recourse has become a popular idea among policymakers, and there is proposed legislation in both the United States and Europe that would mandate designing algorithms that allow recourse for affected persons.

When possible, ideas from **participatory design** should always be used to determine stakeholder goals. Participatory design, also called co-operative design or co-design, is an approach to design wherein those stakeholders identified in Step 2A are *actively involved in the design process* to help ensure the result meets their needs. In one promising example, designers used participatory design to successfully create better explanations about an algorithmic tool in the field of communal energy accounting by having conversations with directly with the tool's users (6).

DELIVERABLE

A list of goals for each stakeholder of each algorithmic decision-making system in your organization. At this point, your running inventory might be quite long – but you can be assured that you have thoughtfully considered all the important aspects of transparency.

Step 2C: Design transparency features for your algorithms given stakeholders and their goals.

Once you have inventoried your list of stakeholders and their needs, you are ready to begin designing transparency features for your algorithmic systems. Importantly, this should be a collaborative design process between **technical** and **non-technical** persons within your organization. Technical experts (like practitioners, data scientists, data engineers, programmers, and analysts) will have additional knowledge on how to implement transparency features (this is further detailed in **Step 2D**).

Importantly, there are two levels of transparency you need to consider, called the **scope of transparency.** The first is **local** transparency, which provides understanding about a single decision made by an algorithm (ex. a single loan applicant), and second is **global** transparency, which explains how an algorithm works overall. Global transparency can give a "bird's-eye view" of an algorithmic decision-making system, whereas local transparency focuses in on a particular bird or set of birds.

There are many types of transparency features you want to consider that extend even beyond the scope of this playbook, but here we detail 5 ideas:

- **Transparency labels** for algorithmic decision-making systems are modeled after the kind of nutritional labels found in the food industry. Nutritional labels are designed to be transparent and understandable, and their contents are perceived as a highly credible source of information by consumers, who use them to guide decision making.

  There are several examples of transparency labels for algorithms that have been designed by researchers, and like nutritional labels, they often contain the "ingredients" that make up an algorithm (26). For example, the labels may include descriptive information about the factors considered by the algorithm (the ingredients), how they are ranked in terms of their importance in the decision-making (ingredient ranking), and attributes related to fairness, which could be useful for meeting stakeholder goals related to validity, trust, privacy, and fairness.

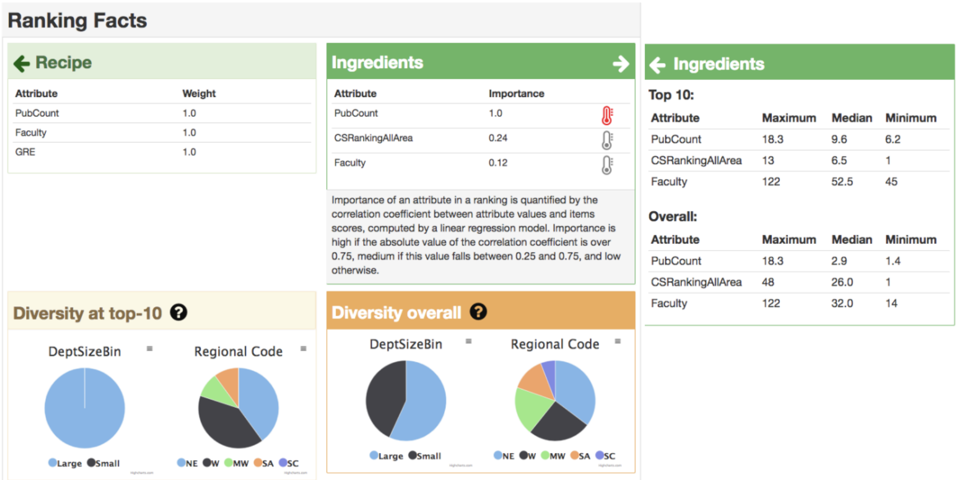

Figure 1: The figure above shows a **transparency label** for an algorithm and comes from the tool *RankingFacts*. It shows the "recipe" for the algorithm (those attributes that are considered and their relative weights), as well additional information about the ingredients. This transparency label also shows measures related to the fairness of the algorithm, like how different subgroups are being classified.

> **When is this useful?**
>
> To provide a global, high-altitude view of the algorithmic system for aspects like (1) what data is being used by the system, (2) how the system weighs that data, (3) metrics on the performance or fairness of the system overall.

- **Data visualizations** can be used to show information about the data used in to create an algorithmic decision system, or facts about the system itself, like how many decisions an algorithm makes per day, and how many people are affected by those decisions.

  Visualizations have proved useful for informing users and making complex information more accessible and digestible, and have even been found to have a powerful persuasive effect (21; 25). Visualizations are often an advisable tool for transparency as they can easily convey lots of information in a simple manner, and organizations commonly staff analysts who specialize in visualizations.

  It's also important that visualizations are designed thoughtfully, as they have the ability to be abused and can successfully misrepresent a message through techniques like exaggeration or understatement (22).

> **When is this useful?**
>
> Useful for presenting complex information in a digestable way, particularly for non-technical users. This could include both internal and external stakeholders, like humans-in-the-loop for the former and affected individauls for the latter.

- Some (but not all) algorithms have built in **intrinsic transparency mechanisms** *(also called intrinsic explainability mechanisms)* that simply need to be surfaced to offer transparency into how they work.

  For example, two common algorithm types are **decision trees** and **rules-lists.** For the former it is possible to print out and display the tree diagram for the user. For the latter, one can list out all the rules used to make a decision for. Another type of commonly used algorithm are **linear classifiers**, which can produce formulas that explain their decision-making. These formulas are sometimes very easy to understand.

Unfortunately, many highly sophisticated algorithms like **random forests** and **neural networks** do not have intrinsic transparency mechanisms. Importantly, the practitioners who designed the algorithm will be aware of whether or not intrinsic transparency mechanisms are available (see **Step 4D**).

> When is this useful?
>
> To answer the question "how does the system work, to the extent that given a new input to the algorithm, I could anticipate the DELIVERABLE with a high degree of accuracy?" Generally for providing a deeper understanding of how the underlying algorithm in the system functions.

- The **attribute importance** *(also called feature importance or factor importance)* of an algorithm is a list that shows all the different attributes (sometimes called features or factors) that are considered by an algorithm, and their relative weights. It offers *global transparency* for an algorithm.

  For example, consider an algorithm that makes predictions on whether or not an individual should receive a loan. The attribute importance could be made up of three attributes: an individual's income, their credit history, and their education level. The weights for these attributes in the algorithm's decision-making may be 40% income, 40% credit history, and 20% education level.

  There are three advantages to using attribute importance. First, attribute importance can be created for any algorithm, no matter how complicated it is. Second, there are a lot of interesting ways to display attribute importance to a human user through data visualizations. Third, from a technical perspective, it is easy to extract the attribute importance from an algorithm.

> When is this useful?
>
> Provides a global understanding of how an algorithmic system is processing data at a slightly deeper level than what is often found in transparency labels. Useful for *learning and support* and to some extent *recourse*. Useful to practitioners for checking the *validity* of an algorithmic system.

- The **attribute influence (also called feature influence or SHAP factors)** of an algorithm is similar to the attribute importance, except that it shows how the attributes of a single instance or individual impacted the algorithm's DELIVERABLE. A SHAP diagram can be seen in the introduciton to this playbook. In contrast to attribute importance, the influence shows the *local transparency* for a particular case. Like with attribute importance, the attribute influence can be created for *any* algorithm.

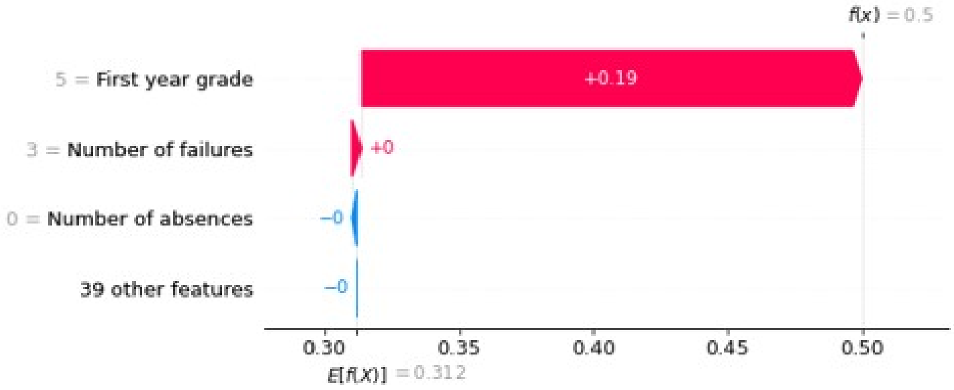

Figure 2: The **SHAP diagram** above was produced for an algorithmic decision system that predicts the likelihood a student will fail out of high school. Each arrow corresponds to an attribute about the student (written on the left side), where red arrows indicate an *increase* in risk of a student failing, while blue arrows indicate a *decrease* in risk. The amount of increase (or decrease) is relative to the size of the arrow. *One can see that the student's first year grade nd high number of failures increases their risk of failure, while the low number of absences slightly decrease their risk.*

Generally, when attribute influence is implemented as a transparency measure for an algorithm, individuals are shown the top 3 to 5 attributes that are influencing the algorithm's DELIVERABLE. Importantly, since attribute influence offers local transparency, it is extremely useful in offering **recourse** to affected persons of an algorithm. It is also very useful for human-in-the-loop users who need transparency for the purposes of decision support.

When is this useful?

To provide local transparency about a single instance, generally an affected individual. The attribute influence is one of the best ways to answer the question, "why did the algorithmic system have this DELIVERABLE for *this specific person*?" Extremely useful for *recourse* for affected individuals. Very useful for *learning and support* by humans-in-the-loop.

Getting Started With Transparency

Given all these options, it can be hard to tell where to start. In fact, because transparency is so intrinsically tied with design, there are no "objectively correct" answers. Research has even revealed some counter-intuitive ideas about transparency, like offering too much information to users can actually confuse them due to information overload (12; 19). This emphasizes the importance of thoughtful design, and when possible, participatory design.

As another guideline, it is not sufficient to try and apply general, one-size-fits-all design like simply implementing a transparency label. First, it is unlikely to achieve all stakeholder goals for transparency. Second, it will likely not be regulatory compliant: both the proposed Algorithmic Accountability Act in the United States and the Artificial Intelligence Act in the European Union specifically mention that algorithmic transparency should allow individuals to have recourse against a system's outcome, which implies the use of local transparency like attribute influence. Many other countries around the world have also begun considering similar legislation.

- **What if I don't know where to begin?**

  As a baseline, if you are unsure which transparency features to choose your algorithm, you should strongly consider implementing transparency labels, attribute importance, and attribute influence for algorithms impacting people and their lives, and tailor them to meet your stakeholders' needs.

- **What is the difference between a good transparency design and a bad transparency design?**

  Unfortunately, there are currently no ways of objectively measuring the quality of transparency in algorithmic decision systems. There is also no research consensus on best practices for transparency. As a result, the quality of algorithmic transparency within your organization is subjective and ultimately up to the algorithm's' stakeholders and whether or not they feel the transparency designs you create meet their transparency goals.

DELIVERABLE

Ideas and designs for which transparency features you want to implement for the algorithms in your organization, and how those features will meet stakeholder goals.

Step 2D: Speak with your technical team to review your design ideas.

After deciding on a set of transparency features to implement, it's important to loop-in your technical team (like practitioners, data scientists, data engineers, programmers, and analysts). In fact, you may want to include them in **Step 2C** if bandwidth allows.

Including your technical team in design discussions is critical for designing and implementing strong transparency measures because of the information asymmetry that exists between what is technically feasible in terms of transparency, and what is ideal for stakeholders. For example, your technical team will be aware of whether or not intrinsic transparency mechanisms exist for your algorithms, or if other transparency tools can be applied to your algorithms.

In general, it is the responsibility of those who design algorithms to understand how they can be made transparent. Luckily, many of those who build algorithms are already aware of transparency features through their experience debugging algorithms and testing their validity.

DELIVERABLE

A refined idea and design for implementing transparency for your organization's algorithms.

STEP 3: Implement transparency into your algorithms

Once you have identified all the algorithms within your organization, their stakeholders, the stakeholders' goals, and created ideas on what transparency features should be implemented, it is finally time to implement them and make your algorithms transparent. This step mainly falls on the responsibility of the technical team, but also includes conversations with stakeholders for testing and feedback purposes.

The Technical Guide

The main contents of this playbook do not provide technical details for implementing transparency features, but they can be found in the Appendix Technical Guide.

Step 3A: Implement transparency features.

Implementing transparency features is a technical process that may include creating data visualizations, building web or mobile dashboards for algorithms, or creating algorithmic factsheets. We have seen two such implementations: the first is the SHAP diagram seen in the introduciton, and the second is the RankingFacts transparency labels seen above.

Here is another example of a dashboard:

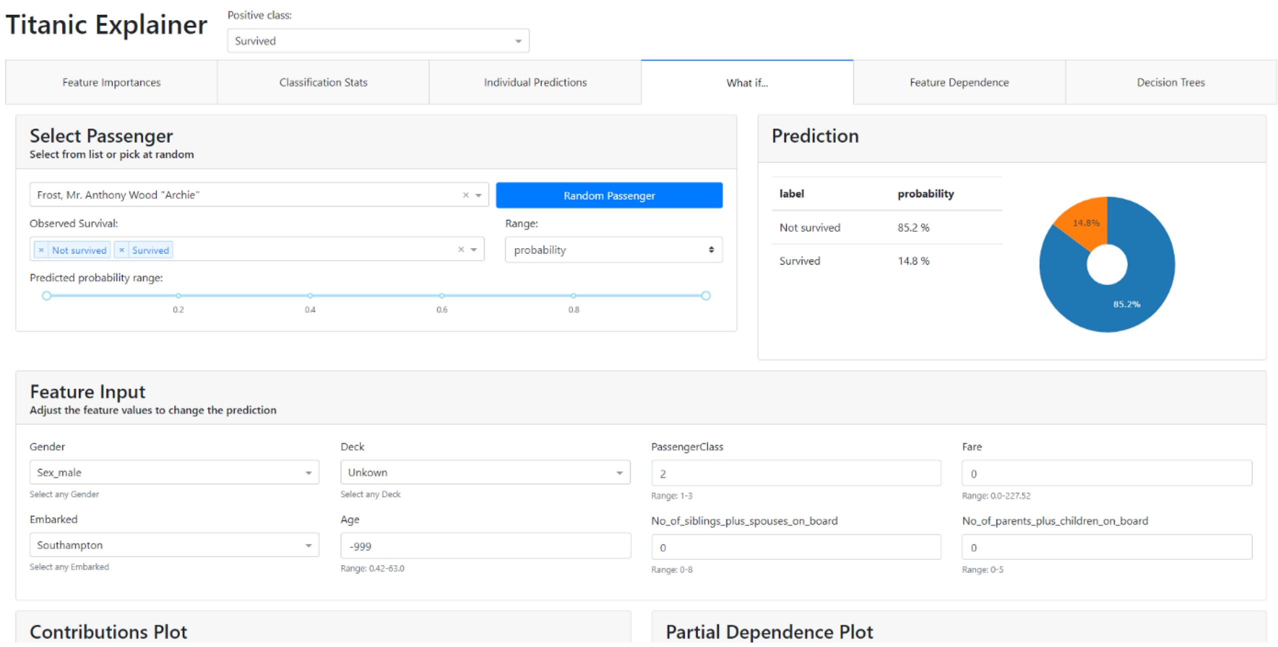

Figure 3: The figure above shows **a dashboard for an algorithm** that predicts who would have survived the *Titanic* crash based on attributes like their gender and the price of their ticket (1st, 2nd, or 3rd class). In the widget titled *Select Passenger*, the user is able to select a particular passenger. In the widget labeled emphPrediction an estimate for how likely it is the selected passenger would have survived. The widget *Feature Input* allows the user to change attributes about the individual (ex. their age to see how it would impact their prediction.

DELIVERABLE

Full implementations of transparency for your organization's algorithms.

Step 3B: Test and review your transparency implementation with stakeholders.

As mentioned earlier, there is no objective way of measuring whether or not transparency has been implemented properly. Instead, the quality of transparency is subjective and ultimately up to the stakeholders. To avoid the pitfalls of poorly implemented transparency features, you should consult with stakeholders to make sure that transparency features are implemented properly. We have included some design tips in the **Appendix** of this playbook.

Some pitfalls of poorly implemented transparency include showing attribute importance and influence in a way that they are not human understandable, assuming that your stakeholders have the same knowledge as you or the technical team (ex. not everyone understands how to read a decision tree diagram), or designing global transparency when local transparency is needed, or vice-versa.

DELIVERABLE

Refined implementations of transparency for your organization's algorithms. You are nearing the finish line!

STEP 4: Maintain your algorithms and transparency

Step 4A: Create a process for continually maintaining algorithmic transparency.

Like with all technical implementations, they need to be maintained and monitored throughout their use. This is particularly true for predictive algorithms, which are subject to a phenomena known as *concept drift*, wherein the accuracy of predictions tend to worsen over time due to attributes becoming less correlated with whatever is being predicted (14). As a consequence of concept drift, explanations may shift or become less meaningful over time.

Furthermore, the interest in algorithmic transparency is growing rapidly, and new methods for algorithmic transparency are being created on a yearly basis. A new breakthrough in transparency could come at any moment – and it's critical that your organization stays up to date with best practices!

With these considerations in mind, we highly recommend you consult with relevant stakeholders, particularly those building algorithmic systems and those using them, to design a plan for maintaining, updating, and repairing the transparency features of algorithmic systems.

DELIVERABLE

A process for continually maintaining algorithmic transparency within your organization.

Case Study: Public Employment

The following case studies are intended to make the lessons of this playbook more concrete. These case studies we present here are fictional, but based on real-world examples.

Context

Algorithmic decision-systems are increasingly being used by government agencies to help unemployed persons. In particular, many governments are concerned with identifying individuals that are at risk of becoming *long-term* unemployed, that is, being unemployed for 12-months or more. The long-term unemployed are particularly vulnerable persons, and tend to earn less once they find new jobs, have poorer health and have children with worse academic performance as compared to those who had continuous employment (20).

In an ideal world, public employment agencies would be able to identify individuals who are likely to be long-term unemployed so that they can be given appropriate interventions to get ahead of the problem, like job search trainings, resume workshops, or reskilling programs. This is where algorithmic decision-systems come in to play: governments can use these systems to spot those at risk of becoming long-term unemployed. For instance, a system of *exactly* this type is used by the national Portuguese vocational agency (27).

How the system works

Once a person becomes unemployed, their change in employment status is often reported to a government agency, either by their previous employer or when an individual applies for unemployment benefits. For example, in the US, you must go through your state-level Department of Labor to receive benefits.

In this scenario, once a person registers that they are unemployed, they are assigned a case-worker known as a "job counsellor." The job counselor will reach out to the individual (either by phone or by scheduling an in-person meeting) to gather information like their age, marital status, education level, and their work history. This information, along with macroeconomic data related to the individual's employment sector and location of work, are passed into an **algorithmic system** used by the job counselor that generates a risk score between 1 and 5 for how likely it is that the person will become long-term unemployed. If someone is identified as a risk level of 5, they will be offered free job-related interventions by the state like resume workshops or even job trainings.

Using the Algorithmic Transparency Checklist

We have already completed Step 1 by identifying the algorithmic decision system being used. We can now complete Step 2a-b, and create a list of stakeholders and their potential goals:

Stakeholder	Goals
Humans-in-the-loop (job counselors)	Learning and support, validity. The job counselor may want to know how the system works overall, and why a particular individual is assigned to a certain risk score.
Affected persons (un-emlpoyed individual)	Recourse, trust. An individual may believe they are high risk and deserve interventions regardless of the output of the algorithm (or vice-versa)
Managers (ex. officials at the Department of Labor)	Learning and support, trust, validity, fairness. The managers may have si-miliar questions to the job counselor and practitioner, plus concerns about the overall fairness of the system.
Compliance officers (ex. auditors)	Fairness, privacy. Since the system is working with sensitive data, auditors will want to make sure the system is compliant with applicable fairness and privacy laws.
Practitioners	Learning and support, validity, trust. Even though the system is already built, practitioners may be interested in monitoring its performance over time to

Ideally, we would begin applying *participatory design* at this point, and speak with all the relevant stakeholders to verify that we understand their transparency goals. This is particularly true for job counselors using the system, and the unemployed persons affected by the outcome.

Here are some examples of what we might hear if we talked with stakeholders:

- **Affected persons.** "I want to know if an algorithmic system is being used on me."; "I would like to know what data is being used in an algorithmic system"; "I want to be able to challenge the outcome of the system."
- **Job counselors.** "I need to understand why the algorithmic system assigns a person to a particular risk score. Based on my experience as a job counselor, I may agree (or disagree) with the system's prediction."
- **Managers.** "I want to understand what data is being used by the algorithmic decision system, and what factors it considers important. I want to have a 'bird's eye view' of the system. I would also like to know if job counselors are consistently agreeing (or disagreeing) with the risk estimates produced by the algorithm."

Next we move to Step 2C and Step 2D, where we design transparency features for our algorithmic system and then speak with the technical team about implementing them. Ideally, we would continue the participatory design process and gather feedback on our transparency features from the respective stakeholders.

We can begin by considering the goals of managers and compliance officers, who are concerend a high-altitude view of the algorithmic system. This is a perfect applciation for transparency labels, which presents snapshots on the data being used by the algorithm, how it rates the importance of that data (the attribute influence), and even metrics related to how the system performs for different protected groups. This could be accompanied by text describing what measures are taken to ensrue the privacy of individuals. Below is a mock-up of transparency labels:

Two other uses of tansparency labels include informing affected individuals on the existence of the algorithm, and for training job counselors about the system.

Next, for learning and support for job counselors and for recourse for individuals, some type of local transparency method must be made visible. A good choice for this is showing the attribute influence for each risk estimate made the system. Also, it may be necessary to contextualize the informaiton shown to users. For example, if an unemployed person has a risk score of 3, what does this *accutally* mean? What is the chance that a person with a risk score of 3 will become long-term unemployed relative to those with higher or lower scores? Two strong technical choices for attribute influence in this scenario would be SHAP explanations or counterfacutal explanations.

Lastly, the technical team and managers within your organization may want a customized dashboard that displays technically-revelant performance metrics about the algorithm. Managers may want to understand how often job counselors agree or disagree with the system, as well as track the overall accuracy of the system in predicting long-term unemployed persons. These types of metrics would be helpful in surfacing degredation of the model over time.

Before moving to Step 3, it is also a could idea to review the *Design Considerations* section of this playbook to make sure you have avoided any transparency pitfalls. One dashboard containing all the information above may very likely be counter productive to transparency by overwhelming stakeholders with too much information that is relevant to their specific goal.

With the design work out of the way, it is time to move to Step 3A and Step 3B where the technical team implements the transparency solutions (they may have to build several dashboards as described above) and tests them with the end users.

At this point, we are finished implementing transparency into the system! All we need to do now is move on to Step 4A and make sure a process is in place to continually monitor the system and the transparency features into the future.

Acknowledgements

The research underpinning this document was supported in part by NSF Awards No. 1934464, 1922658, 1916505, 1928614, and 2129076.

The accompanying course for this playbook was published in the 2023 *ACM CHI Conference on Human Factors in Computing Systems* proceedings, and can be accessed here[1] (3). A pre-cursor to this work was also published in *Data & Policy* (4).

The authors thank the following list of contributors for generously providing input and advice in the development of this playbook:

- Taka Ariga, Chief Data Scientist and Director of Innovation Lab, U.S. Government Accountability Office
- Nicholas Bell, Ph.D.
- John Rood, Founding Account Executive, Proceptual
- Ken Hellberg, Founding Account Executive, Proceptual
- Emily McReynolds, NYU Center for Responsible AI

Appendix

Design Guide

Know your needs

Algorithmic transparency is always for human users. To this end, best design practices should include ideas from the fields like *participatory design (PD)* and *human-computer interaction (HCI)*. Earlier in this playbook, we mentioned several key concepts from these fields, like including stakeholders in the ideation and design process. We also stressed the importance of asking questions like who and what are you designing for? In other words, you must know your transparency needs before beginning the design process.

For purposes of scope, we will not delve deeper into PD and HCI in this playbook—but if you would like to learn more, we recommend the following reading:

- *Configuring participation: on how we involve people in design* by John Vines, Rachel Clarke, Peter Wright, John McCarthy, and Patrick Olivier
- *The UX Book: Designing a Quality User Experience* by Pardha S. Pyla and Rex Hartson
- Liberating Structures, an interactive website that presents activities for PD.

Design Considerations

Below we list several design considerations that are specific to algorithmic transparency, along with *key questions* you can ask yourself to make sure your transparency complies with this design.

- **Make Explanations About a Decision Useful and Actionable**. Not all explanations have the same utility for stakeholder goals. This is particularly true for transparency mechanisms that are implemented for the purposes of of **recourse** or redress by an affected individual against an algorithm's output (2).

[1] https://dataresponsibly.github.io/algorithmic-transparency-playbook/

For example, consider an algorithm that determines whether or not an applicant will be accepted or rejected for a loan based on factors like income, age, and credit score. If an applicant is denied a loan and explanations are generated automatically by SHAP, it may tell the individual that the most important feature impacting the decision was their age – which is something that the individual can do nothing about. Similarly, a counterfactual explanation telling the applicant that they must increase their income 10-fold may be comparatively much more difficult than a small improvement in credit score (13).

Key question: is the way transparency has been implemented actually useful in meeting the goals of the stakeholders? Note that involving stakeholders in the design process may be critical to avoid implementing useless and in-actionable explanations.

- **Less May be More.** One's initial instinct when implementing transparency for automated decision systems is to provide as much information as possible about the system. While it's important to be open about many aspects of automated decision systems (especially with respect to how fair or trustworthy they are), overloading stakeholders with information may actually have the counterintuitive effect of making a system seem *more* opaque. This is not always the case, but there are notable research studies showing negative impacts on the perceived understanding of decision systems by users due to information overload (5).

  Key question: has too much transparency been implemented into the system in such a way that stakeholders may be confused or misled? Note that involving stakeholders in the design process may be critical to avoid implementing useless and inactionable explanations.

- **Don't Manipulate Users.** While it should go without saying, transparency mechanisms should not be implemented in a way that deceives or manipulates the stakeholders of automated decision systems. Researchers have uncovered "*dark patterns*" of transparency that can create a false sense of security for users, and trick them into believing the system is trustworthy or fair when the underlying model is biased against minority groups.

  Here are two more examples of dark patterns: first, in one context, researchers found that giving users large volumes of information may arbitrarily make a model appear more fair or trustworthy, even when the additional information has nothing at all to do with the relative fairness of the model (23). Second, research has shown that data visualizations can successfully be used to misrepresent a message through techniques like exaggeration or understatement (8).

  A taxonomy of dark patterns can be found here. It's important for those implementing transparency to be aware of dark patterns and pitfalls in transparency so they can be audited for and removed from their work.

  Key question: is the way transparency has been implemented fair, honest, and ethical?

- **Consider the Performance-Use Paradox.** The performance-use paradox is a phenomena that was observed for an automated decision system implemented in a public employment setting wherein the users of the system claimed that while they rarely used the output and explanations generated by the system, they preferred having the system as opposed to having it removed. The reason for this phenomena was that users felt more confident in their own decision-making, but liked having the system's output and explanation as a "potential backup" (27).

  For designers of transparent automated decision systems, the performance-use paradox provides an important lesson in understanding that the system may not be the primary means for decision-making, but at worst should be providing explanations that can be used to support and back-up decisions made by human users.

Key question: is the way transparency has been implemented supporting decision-making in a way that users will be comfortable having the system as a potential backup?

- **Transparency is Not Inherent to any System or Algorithm.** Many practitioners and researchers in the machine learning community have created a list of algorithm types that are commonly accepted as being transparent (also called "interpretable"), which consists of linear models, decision trees, and rules-based models. These algorithms are those with *intrinsic transparency mechanisms*, like their linear formula, tree diagram, and rules-list, respectively.

  However, recent research has called into question the inherent transparency of these algorithms. For example, it was shown that tree diagrams offer very poor transparency when it comes to having stakeholders identify the most important attribute used in the system's decision process (5).

  Furthermore, it is important to consider the complexity of these algorithms. If a linear model or rules-list is made up of hundreds or thousands (or more) of rules it will likely no longer be inherently transparent to stakeholders. In fact, one could make an argument that a small neural network with only a few nodes may be more transparent than large rules-lists.

  The implication of this design principle is that designers must always consider the value of additional transparency mechanisms even when working with simple algorithms like linear models, decision trees, and rules-based models.

  Key question: is the way transparency has been implemented robust beyond intrinsic transparency mechanisms?

Technical Guide: A Survey of Transparency Methods

There are a wide range of transparency tools available – some of which have been around well before the resurgence of AI, and some developed in recent years. Here we catalog a number of known methods, provide a short several details relevant to their implementation, and links to relevant documentation (when applicable). Note that the XAI field is in constant flux and new methods for explainability are being developed on a near-yearly basis.

Intrinsic explainability mechanisms

Some algorithms have built-in tools for explainability. These include algorithms like decision trees, linear models, and rules-list where artifacts like the tree diagram, the linear formula, and a list of if-then rules, respectively can be easily extracted from the model. Researchers have also developed experimental methods for extracting intrinsic explainablity from black-box. One such example is Self-Explaining Neural Networks.

Counterfactual explanations

A full discussion of counterfactual explanations can be found here, but in summary, counterfactual explanations are local example-based explanations that are used to show how changes in input features impact outputs.

For counterfactual explanations of a single individual, features are perterbutated to see the impact on the outcome. For example, one can modify an individual's income or education level to understand how that would influence an algorithm's decision to grant or deny them a loan.

Post-hoc explainability methods

Post-hoc explainability methods can be very useful for creating explanations of both black-box and interpretable models. These methods generate local explanations about an input into an already trained algorithm or model (hence "post-hoc"). In order of popularity, the most commonly used are SHAP, LIME, and QII. There is also a new method known as SAGE by the creators of SHAP that provides global model explanations.

In industry, SHAP is the most commonly used post-hoc explainability method. This is because it is intuitive and easy to implement due its well-maintained Python package. An implementation of SHAP could include showing stakeholders the top 3 factors that influence their output positively, and the 3 factors that influence their output negatively.

While there are many positives to post-hoc explainability methods like SHAP (intuitive, easy to understand), there are several weaknesses that should be noted: (1) there are several examples of instability in post-hoc explanations where similar inputs yield very different explanations, (2) features identified by SHAP as important are not necessarily those which are actionable, important, or meaningful to stakeholders (3) post-hoc explainability methods are vulnerable to adversarial attacks (24), (4) the built-in visualizations created by SHAP were designed for data scientists and are not always useful or easily understood by other stakeholders.

Crosstabs

Crosstabs are a model agnostic method that provide global or group-level explanations about the outputs of a model. Crosstabs are simply summaries of the top 10-25% highest (or lowest) scored outputs of an algorithm.

For example, crosstabs for an algorithm used to predict whether or not an individual will be approved for a loan may show that the 10% most likely to be approved have an average income of $100,000 and a credit score of 750, versus the 10% most likely to be denied which may have an average income of $15,000 and a credit score of 350.

One advantage of crosstabs is that they can be used to provide group level explanations. One such use-case is comparing crosstabs between members of different protected groups to understand the fairness of your algorithm.

Simplifying algorithms

One technique that can improve the explainability of complex algorithms is replacing them with simpler models that make similar predictions. In many cases (but not all), simpler models can be used to approximate the outputs made by more complex models. To do this, create a simpler model and compare the outputs of that model using Jaccard plots of similarity. Indications of high similarity may show that you can drastically reduce the feature space of a model while maintaining high fidelity to the models original output.

There are also methods for directly reducing the complexity of models, like select-regress-round where complex linear models are reduced to short rules-list (10). Empirical evidence shows that in many cases using select-regress-round does not significantly impact the accuracy of an algorithm.

References

[1] Kasun Amarasinghe, Kit T. Rodolfa, Hemank Lamba, and Rayid Ghani. 2020. Explainable Machine Learning for Public Policy: Use Cases, Gaps, and Research Directions. *CoRR* abs/2010.14374 (2020). arXiv:2010.14374 https://arxiv.org/abs/2010.14374

[2] Solon Barocas, Andrew D Selbst, and Manish Raghavan. 2020. The hidden assumptions behind counterfactual explanations and principal reasons. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 80–89.

[3] Andrew Bell, Oded Nov, and Julia Stoyanovich. 2023. The Algorithmic Transparency Playbook: A Stakeholder-First Approach to Creating Transparency for Your Organization's Algorithms. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI EA '23)*. Association for Computing Machinery, New York, NY, USA, Article 554, 4 pages. https://doi.org/10.1145/3544549.3574169

[4] Andrew Bell, Oded Nov, and Julia Stoyanovich. 2023. Think about the stakeholders first! Toward an algorithmic transparency playbook for regulatory compliance. *Data & Policy* 5 (2023), e12.

[5] Andrew Bell, Ian Solano-Kamaiko, Oded Nov, and Julia Stoyanovich. 2022. It's Just Not That Simple: An Empirical Study of the Accuracy-Explainability Trade-off in Machine Learning for Public Policy. In *2022 ACM Conference on Fairness, Accountability, and Transparency*. 248–266.

[6] Florian Cech. 2021. Tackling Algorithmic Transparency in Communal Energy Accounting through Participatory Design. In *C&T'21: Proceedings of the 10th International Conference on Communities & Technologies-Wicked Problems in the Age of Tech*. 258–268.

[7] Jeffrey Dastin. 2018. Amazon scraps secret AI recruiting tool that showed bias against women. In *Ethics of Data and Analytics*. Auerbach Publications, 296–299.

[8] Upol Ehsan and Mark O Riedl. 2021. Explainability pitfalls: Beyond dark patterns in explainable AI. *arXiv preprint arXiv:2109.12480* (2021).

[9] Anna Jobin, Marcello Ienca, and Effy Vayena. 2019. The global landscape of AI ethics guidelines. *Nature Machine Intelligence* 1, 9 (2019), 389–399.

[10] Jongbin Jung, Connor Concannon, Ravi Shroff, Sharad Goel, and Daniel G Goldstein. 2017. Simple rules for complex decisions. *arXiv preprint arXiv:1702.04690* (2017).

[11] Andrei Kirilenko, Albert S Kyle, Mehrdad Samadi, and Tugkan Tuzun. 2017. The flash crash: High-frequency trading in an electronic market. *The Journal of Finance* 72, 3 (2017), 967–998.

[12] Q. Vera Liao, Daniel M. Gruen, and Sarah Miller. 2020. Questioning the AI: Informing Design Practices for Explainable AI User Experiences. In *CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, April 25-30, 2020*, Regina Bernhaupt, Florian 'Floyd' Mueller, David Verweij, Josh Andres, Joanna McGrenere, Andy Cockburn, Ignacio Avellino, Alix Goguey, Pernille Bjøn, Shengdong Zhao, Briane Paul Samson, and Rafal Kocielnik (Eds.). ACM, 1–15. https://doi.org/10.1145/3313831.3376590

[13] Q Vera Liao and Kush R Varshney. 2021. Human-centered explainable ai (xai): From algorithms to user experiences. *arXiv preprint arXiv:2110.10790* (2021).

[14] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. 2018. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering* 31, 12 (2018), 2346–2363.

[15] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 4765–4774.

[16] Ricards Marcinkevics and Julia E. Vogt. 2020. Interpretability and Explainability: A Machine Learning Zoo Mini-tour. *CoRR* abs/2012.01805 (2020). arXiv:2012.01805 https://arxiv.org/abs/2012.01805

[17] Christian Meske, Enrico Bunde, Johannes Schneider, and Martin Gersch. 2020. Explainable Artificial Intelligence: Objectives, Stakeholders and Future Research Opportunities. *Information Systems Management* (12 2020). https://doi.org/10.1080/10580530.2020.1849465

[18] Marcia K Meyers, Susan Vorsanger, B Guy Peters, and Jon Pierre. 2007. Street-level bureaucrats and the implementation of public policy. *The handbook of public administration* (2007), 153–163.

[19] Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. 2018. How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1802.00682* (2018).

[20] Austin Nichols, Josh Mitchell, and Stephan Lindner. 2013. Consequences of long-term unemployment. *Washington, DC: The Urban Institute* (2013).

[21] Anshul Vikram Pandey, Anjali Manivannan, Oded Nov, Margaret Satterthwaite, and Enrico Bertini. 2014. The Persuasive Power of Data Visualization. *IEEE Trans. Vis. Comput. Graph.* 20, 12 (2014), 2211–2220. https://doi.org/10.1109/TVCG.2014.2346419

[22] Anshul Vikram Pandey, Katharina Rall, Margaret L Satterthwaite, Oded Nov, and Enrico Bertini. 2015. How deceptive are deceptive visualizations? An empirical analysis of common distortion techniques. In *Proceedings of the 33rd annual acm conference on human factors in computing systems*. 1469–1478.

[23] Jakob Schoeffer, Niklas Kuehl, and Yvette Machowski. 2022. " There Is Not Enough Information": On the Effects of Explanations on Perceptions of Informational Fairness and Trustworthiness in Automated Decision-Making. *arXiv preprint arXiv:2205.05758* (2022).

[24] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2020. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 180–186.

[25] Aner Tal and Brian Wansink. 2016. Blinded with science: Trivial graphs and formulas increase ad persuasiveness and belief in product efficacy. *Public Understanding of Science* 25, 1 (2016), 117–125.

[26] Ke Yang, Julia Stoyanovich, Abolfazl Asudeh, Bill Howe, HV Jagadish, and Gerome Miklau. 2018. A nutritional label for rankings. In *Proceedings of the 2018 international conference on management of data*. 1773–1776.

[27] Leid Zejnilović, Susana Lavado, Íñigo Martínez de Rituerto de Troya, Samantha Sim, and Andrew Bell. 2020. Algorithmic Long-Term Unemployment Risk Assessment in Use: Counselors' Perceptions and Use Practices. *Global Perspectives* 1, 1 (2020).