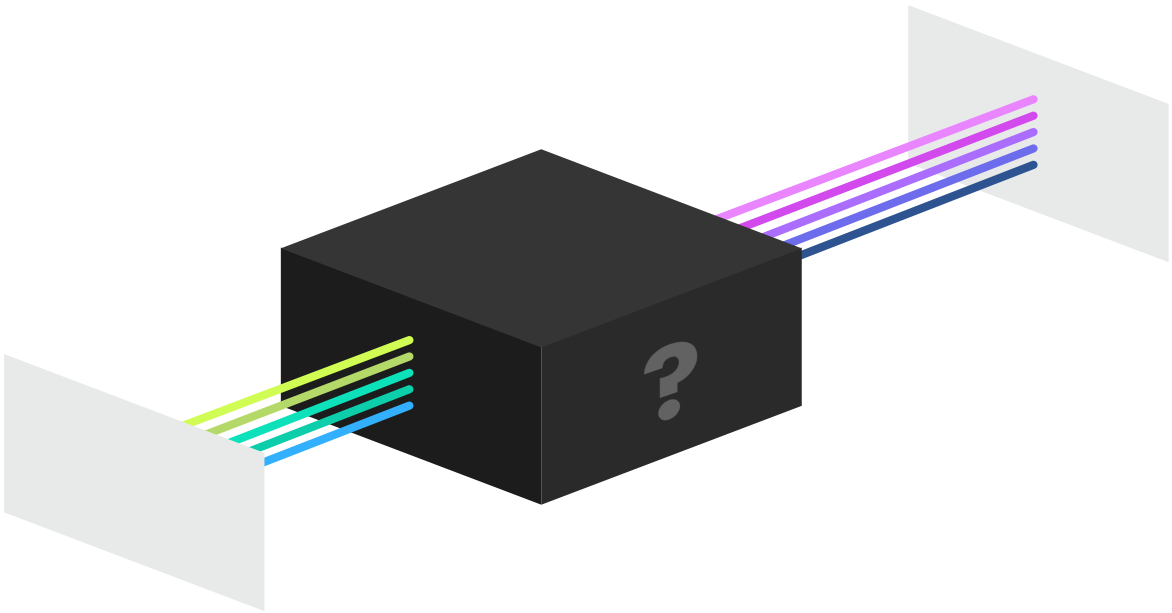
Beyond Explainability: A Practical Guide to Managing Risk in Machine Learning Models



Andrew Burt
Chief Privacy Officer and
Legal Engineer, Immuta

Stuart Shirrell
Legal Engineer,
Immuta

Brenda Leong
Senior Counsel and Director of
Strategy, Future of Privacy Forum

Xiangnong (George) Wang
2018 Immuta Scholar;
J.D. Candidate, Yale Law School

How can we govern a technology its creators can't fully explain?

This is the fundamental question raised by the increasing use of machine learning (ML)—a question that is quickly becoming one of the biggest challenges for data-driven organizations, data scientists, and legal personnel around the world.[1] This challenge arises in various forms, and has been described in various ways by practitioners and academics alike, but all relate to the basic ability to assert a causal connection between inputs to models and how that input data impacts model output.

According to Bain & Company, investments in automation in the US alone will approach $8 trillion in the coming years, many premised on recent advances in ML.[2] But these advances have far outpaced the legal and ethical frameworks for managing this technology. There is simply no commonly agreed upon framework for governing the risks—legal, reputational, ethical, and more—associated with ML.

This short white paper aims to provide a template for effectively managing this risk in practice, with the goal of providing lawyers, compliance personnel, data scientists, and engineers a framework to safely create, deploy, and maintain ML, and to enable effective communication between these distinct organizational perspectives. The ultimate aim of this paper is to enable data science and compliance teams to create better, more accurate, and more compliant ML models.[3]

Does It Matter How Black the "Black Box" Model Is?

Many of the most powerful ML models are commonly referred to as "black boxes," due to the inherent difficulty in interpreting how or why the models arrive at particular results. This trait is varyingly addressed as "uninterpretability," "unexplainability," or "opacity" in the legal and technical literature on ML. But a model's perceived opacity is often the result of a human decision: the choice of which type of ML model to apply. Predictive accuracy and explainability are frequently subject to a trade-off; higher levels of accuracy may be achieved, but at the cost of decreased levels of explainability.[4]

While limitations on literal explainability are a central, fundamental challenge in governing ML, we recommend that data scientists and lawyers *document this trade off from the start*, due to the fact that there are various ways to balance accuracy against explainability. Data scientists might seek to break down the decision they're predicting using ensemble methods, for example, utilizing multiple models to maximize accuracy where necessary while maximizing explainability in other areas. Any decrease in explainability should always be the result of a conscious decision, rather than the result of a reflexive desire to maximize accuracy. All such decisions, including the design, theory, and logic underlying the models, should be documented as well.[5]

Similarly, we recommend *all lines of defense take into account the "materiality" of the model deployment*. Broadly speaking, the concept of materiality arises from evaluating the significance or personal impact of the model on the organization, its end users, individuals, and third parties. In practice, for example, a model making movie recommendations will have lower impact—and therefore should allow a higher tolerance for unknown risks—than a model used in a medical environment, the results of which could have a direct impact on patient health.

Key Objectives & The Three Lines of Defense

Projects that involve ML will be on the strongest footing with clear objectives from the start. To that end, all ML projects should begin with *clearly documented initial objectives and underlying assumptions*. These objectives should also include major desired and undesired outcomes and should be circulated amongst all key stakeholders. Data scientists, for example, might be best positioned to describe key desired outcomes, while legal personnel might describe specific undesired outcomes that could give rise to legal liability. Such outcomes, including clear boundaries for appropriate use cases, should be made obvious from the outset of any ML project. Additionally, expected consumers of the model—from individuals to systems that employ its recommendations—should be clearly specified as well.[6]

Once the overall objectives are clear, *the three "lines of defense" should be clearly set forth*. Lines of defense—inspired by model risk management frameworks like the Federal Reserve Board's Supervision and Regulation Letter 11-7—refer to roles and responsibilities of data scientists and others involved in the process of creating, deploying, and auditing ML. SR 11-7, for example, stresses the importance of "effective challenge" throughout the model lifecycle by multiple parties as a crucial step that must be distinct from model development itself.[7] The ultimate goal of these measures is to develop processes that direct multiple tiers of personnel to assess models and ensure their safety and security over time. Broadly speaking, the first line is focused on the development and testing of models, the second line on model validation and legal and data review, and the third line on periodic auditing over time. Lines of defense should be composed of the following five roles:

- **Data Owners:** Responsible for the data used by the models, often referred to as "database administrators," "data engineers," or "data stewards."
- **Data Scientists:** Create and maintain models.
- **Domain Experts:** Possess subject matter expertise about the problem the model is being used to solve, also known as "business owners."
- **Validators:** Review and approve the work created by both data owners and data scientists, with a focus on technical accuracy. Oftentimes, validators are data scientists who are not associated with the specific model or project at hand.
- **Governance Personnel:** Review and approve the work created by both data owners and data scientists, with a focus on legal risk.

Some organizations rely on model governance committees—which represent a range of stakeholders impacted by the deployment of a particular model—to ensure members of each above group performs their responsibilities, and that appropriate lines of defense are put in place before any model is deployed.[8] While helpful, such review boards may also stand in the way of efficient and scalable production. As a result, *executive-led model review boards should shift their focus to developing and implementing processes surrounding the roles and responsibilities of each above group*. These boards should formulate and review such processes before they are carried out and in periodic post-hoc audits, rather than individually reviewing each model before deployment.

We make further recommendations below as to how to develop these three lines of defense. Critically, these recommendations should be implemented in varying degrees, consistent with the overall risk associated with each model. Every model has unforeseen risks, but some deployments are more likely to demonstrate bias and result in adverse consequences than others. As a result, we recommend that the depth, intensity, and

frequency of review factor in characteristics including: the model's intended use and any restrictions on use (such as consumer opt out requirements), the model's potential impact on individual rights, the maturity of the model, the quality of the training data, the level of explainability, and the predicted quality of testing and review.[9]

Implementing the Three Lines of Defense

A select group of data owners and data scientists comprise the first line of defense, documenting objectives and assumptions behind a particular ML project. Another group of data scientists, designated as validators, serves as the second line, along with legal personnel, who together review data quality assessments of the data used by the model, model documentation, key assumptions, and methodologies. *It's critical that data scientists in the second line also serve in the first line in other projects*, in order to ensure that expertise is sufficiently distributed. A third line of defense includes periodic reviews of the underlying assumptions behind the model, including the recommendations below. We recommend third line reviews no less-frequently than every six months. These reviews, however, should be tailored to the specific risks of the ML in deployment, and to the specific compliance burden as well.[10]

Focusing on the Input Data

Once proper roles and processes have been put in place, there is no more important aspect to risk management than understanding the data being used by the model, both during training and deployment. In practice, maintaining this data infrastructure—the pipeline from the data to the model—is one of the most critical, and also the most overlooked, aspects of governing ML.[11] Broadly speaking, effective risk management of the underlying data should build upon the following recommendations:

- **Document Model Requirements:** All models have requirements—from freshness of data, to specific features required, to intended uses, and more—which can impact model performance, all of which need to be documented clearly.[12] This enables validators to properly review each project and ensure that models can be maintained over time and across personnel. Similarly, data dependencies will inevitably exist in surrounding systems that feed data into the model; where these dependencies exist, they should be documented and monitored. Additionally, documentation should include discussion of where personally identifiable information is included and why, how that data has been protected (through encryption, hashing, or otherwise), along with the traceability of that data.

- **Assess Data Quality:** Understanding the quality of data fed into a model is a key component of model risk, and should include an analysis of: completeness, accuracy, consistency, timeliness, duplication, validity, availability, and provenance. Many risk management frameworks rely on the so-called "traffic light system" for this type of assessment, which utilizes red, amber, and green colors to create a visual dashboard to represent such assessments.

- **Encapsulate the Model:** Separating the model from the underlying infrastructure allows for vigorous testing of the model itself and the surrounding processes. To that end, each step—from configuration, to feature extraction, to serving infrastructure, and more—should be clearly documented, and clearly encapsulated, so that debugging and updates can occur without too much complexity. Typically, this complexity accrues with time over the deployment cycle of a model, and is one of the greatest sources of risk in using ML.

- **Monitor the Underlying Data:** Input data should be monitored to detect "data drift," in which production data differs from training data, with an emphasis on how such drift might impact model performance. Data used to train the model should be statistically represented, and data ingested during deployment should be compared against this representation. Thorough leave-one-feature-out evaluations of the model—which can highlight the most determinative features in the underlying data—should also be performed.[13] These evaluations can be used to understand whether specific features in the data should be monitored with extra care, along with potentially underutilized features, which the model may not need to ingest.

- **Make Alerts Actionable:** Monitoring underlying data allows for the detection of potential undesired changes in model behavior—but monitoring is only as useful as the existing alert system.[14] We recommend alerts notify both the data owner and the data scientists in the first line of defense, and that all alerts are saved for logging purposes so the second and third line reviewers can audit how alerts were generated and how they were responded to over time.

Using Model Output Data as a Window into Your Model

Understanding the outputs of a model—both during training and once in deployment—is critical to monitoring its health and any associated risks. To that end, we recommend that data owners, data scientists, validators, and governance personnel:

- **Expose Biases:** Data can inaccurately represent the real world, such as when a dataset omits or isolates a fraction of the population in a systematic way. Data can also reflect socially-derived artefacts in ways that are detrimental to particular groups. As such, removing bias from a model is not always practical, but seeking to quantify that bias—and where possible, to minimize it—is. For data on human subjects, it may be possible to validate outputs by cross-referencing privately-held datasets with public information, such as from a national statistics bureau. Where such validation is not feasible, policies applied to data may also need to restrict sensitive data (such as data on race or gender), and output analysis should be performed to detect potential proxies for sensitive features (such as zip codes).[15] We recommend perturbing sensitive features in input data and using the resulting model output to determine the model's reliance on these sensitive features, in addition to detecting the existence of any features that are acting as proxies (such as age).[16] In practice, detecting bias calls for a mixture of data analysis focused on both model inputs and outputs. Evaluation for bias should occur at all stages of model design and implementation, and throughout each line of defense.

- **Monitor Continuously:** We recommend that model output be statistically represented, just like the underlying training and deployment data the models ingest. This will require a clear understanding of where each model's decisions are stored and establishing a statistical "ground truth" of correct behavior during training. In some cases, these representations will enable anomaly detection, or model misbehavior, to be uncovered in a timely manner. These representations will also help detect whether the input data has strayed from the training data, and can indicate when a model should be retrained on a refreshed dataset. The full impact of these methods will vary—depending, for example, on whether the model continues to train during deployment, among many other factors—but they will enable quicker risk assessment, debugging, and more meaningful alerts.[17]

- **Detect Feedback Loops:** Feedback loops occur when a model's actions influence the data it uses to update its parameters. This could occur, for example, when a content-selection system and an ad-selection system exist on the same page, but do not share parameters and were not jointly trained. The two selection systems can influence one another over time, especially if both are continually updating their internal parameters. Detecting such feedback loops can be challenging and time-consuming. Organizations deploying multiple models that might interact with each other over time should pay particular attention to this phenomenon when monitoring model output.

- **Document All Testing:** All such analysis and testing, especially testing focused on bias within the model, should be clearly documented—both to serve as proof of attempts to minimize or avoid undesired outcomes, and to help members of the second and third lines of defense evaluate and understand the project's development and potential risks. We recommend that testing documentation specify, at minimum, who conducted the testing, the nature of the tests, the review and response process, and delineate the stages at which testing occurred. Critically, all such documentation should be easily available to every member of the first, second, and third line of defense. Making this documentation easily accessible will help ensure that testing is thorough and will enable everyone involved in the model's deployment to clearly understand the associated risks.

As with the above recommendations on underlying data shift, actionable alerts should also be a priority in monitoring the model's output. It is critical that these alerts are received by the right personnel, and that such alerts be saved for auditing purposes.

Pulling Models from Production

There are a variety of reasons why models can and should be removed from a production environment—or simply overridden or corrected. These processes need to be considered at every step of the deployment cycle. Every member of the first, second, and third lines of defense, for example, should *understand how to remove the model from production, what, if any, replacement would be required, and the impact of doing so in the short- and medium-term*.[18] These requirements should be included in the model documentation and should be validated and reviewed for accuracy and consistency over time.[19]

Risk Management: A Never-Ending Task

Effective ML risk management is a continuous process. While this paper has been focused on the deployment of an individual model, multiple models may be deployed at once in practice, or the same team may be responsible for multiple models in production, all in various stages. As such, *it is critical to have a model inventory that's easily accessible to all relevant personnel*. Changes to models or underlying data or infrastructure, which commonly occur over time, should also be easily discoverable. Some changes should generate specific alerts, as discussed above.

There is no point in time in the process of creating, testing, deploying, and auditing production ML where a model can be "certified" as being free from risk. There are, however, a host of methods to thoroughly document and monitor ML throughout its lifecycle to keep risk manageable, and to enable organizations to respond to fluctuations in the factors that affect this risk.

To be successful, organizations will need to ensure all internal stakeholders are aware and engaged throughout the entire lifecycle of the model. This whitepaper aimed to outline a framework for managing these risks, and we welcome suggestions or comments to improve this framework. Please reach out to governance@immuta.com with feedback.

Model Management Checklist: A Starting Guide

1st Line of Defense

- ❒ Who is a member of this team?
- ❒ What is each team member's responsibility?
- ❒ Where will documentation for your project be stored?
- ❒ How will it be accessed, and can every team member, in every line of defense, access this documentation?
- ❒ What are key objectives for this project?
- ❒ What outcomes and key risks need to be avoided or minimized?
- ❒ What are key assumptions behind the project?
- ❒ What are key methodologies behind the project?
- ❒ What are key dependencies, and how have they been taken into account?
- ❒ How can the dataset used to train this model be accessed or recreated?
- ❒ What alternative models were considered?
- ❒ What research or references influenced this project?
- ❒ Was a tradeoff made between accuracy and explainability, and if so, how?
- ❒ Can the 1st line of defense replicate the model as it now exists? What about the model when it was initially deployed?
- ❒ Who comprises the 2nd and 3rd lines of defense?
- ❒ Are members of the 2nd and 3rd lines of defense aware of this project? If not, do they need to be?
- ❒ What is the expected lifecycle for this model?
- ❒ How should this model be removed from production or corrected, if necessary?

2nd Line of Defense

- ❒ Who is a member of this team?
- ❒ What is each team member's responsibility?
- ❒ Were the right policies put in place on the data used to create this model?
- ❒ Are the right policies in place on the data the model will ingest once in production?
- ❒ Is the legal analysis behind this project correct, and has it been implemented properly?
- ❒ Are there risks the 1st line of defense missed?
- ❒ Is the technical analysis behind this project sound, and has it been implemented properly?
- ❒ Are there technical reasons this model might not perform as expected?
- ❒ Are there reasons to believe the quality of the training data or the deployment data will be poor or insufficient?
- ❒ If the model changes for any reason once validated, is it clear how the 2nd line will be notified?

3rd Line of Defense

- ❒ Who is a member of this team?
- ❒ What is each team member's responsibility?
- ❒ How often will review take place?
- ❒ Where is there an inventory of all models in the organization?
- ❒ Where is there documentation for all models in the organization?
- ❒ Are there any specific triggers that could or should mandate review?
- ❒ Are there reasons to believe the technical work behind this model is insufficient?
- ❒ Are there reasons to believe the legal risk assessment behind this model is insufficient?
- ❒ Are there any risks that were not accounted for, either because they weren't addressed or because such risk didn't previously exist?
- ❒ Do you have sufficient authority to pull a model from production if it is found to be overly risky, have technical problems, or has other issues?

Immuta is the fastest way for algorithm-driven enterprises to accelerate the development and control of machine learning and advanced analytics. The company's hyperscale data management platform provides data scientists with rapid, personalized data access to dramatically improve the creation, deployment, and auditability of machine learning and AI. Founded in 2014, Immuta is headquartered in College Park, Maryland. For more information, visit www.immuta.com.

Future of Privacy Forum is a nonprofit organization that serves as a catalyst for privacy leadership and scholarship, advancing principled data practices in support of emerging technologies. FPF brings together industry, academics, consumer advocates, and other thought leaders to explore the challenges posed by technological innovation and develop privacy protections, ethical norms, and workable business practices. For more information, visit www.fpf.org.

References

[1] While the literature on explainability in ML is vast, Zachary C. Lipton provides a thorough overview in "The Mythos of Interpretability," presented at 2016 ICML Workshop on Human Interpretability in Machine Learning, available at https://arxiv.org/abs/1606.03490. Other reviews that address privacy and legal considerations include Joshua Kroll, Joanna Huey, Solon Barocas, Edward Felten, Joel Reidenberg, David Robinson, and Harlan Yu in "Accountable Algorithms," Univ. of Penn Law Review, 2017, available at https://scholarship.law.upenn.edu/cgi/viewcontent.cgi?article=9570&context=penn_law_review, and Paul Ohm and David Lehr, "Playing with the Data: What Legal Scholars Should Learn About Machine Learning," Univ. of CA, Davis Law Review, 2017, available at https://lawreview.law.ucdavis.edu/issues/51/2/Symposium/51-2_Lehr_Ohm.pdf.

We are also aware that this is a growing research area, with more advances in explainability by the day. As such, the difficulties presented by unexplainable models may decrease over time, or so we hope. That being said, we believe that the promise of ML models is directly connected to their inability to be fully explained in logical, human-understandable terms. And that's without focusing on the crucial distinction between models that are fully developed once deployed and models that continue to train while they're exposed to live input data. The problem of explainability, in short, will not disappear in the realm of ML, as we note above.

For a good window into recent advances in model interpretability, we recommend Michael Harradon, Jeff Druce, and Brian Ruttenberg, "Causal Learning and Explanation of Deep Neural Networks via Autoencoded Activations," February 2018, available at https://arxiv.org/pdf/1802.00541.pdf, and Mike Ananny and Kate Crawford, "Seeing Without Knowing: Limits of the Transparency Ideal and its Application to Algorithmic Accountability," New Media and Society, December 2016, available at http://journals.sagepub.com/doi/pdf/10.1177/1461444816676645.

[2] Karen Harris, Austin Kimson, and Andrew Schwedel, "Labor 2030: The Collision of Demographics, Automation and Inequality," Bain & Company Report, February 7, 2018 available at http://www.bain.com/publications/articles/labor-2030-the-collision-of-demographics-automation-and-inequality.aspx.

[3] Note that recommendations here are drawn from lessons learned in implementing model risk management frameworks, such as the Federal Reserve Board's Supervision and Regulation Letter 11-7, the EU Central Bank's guide to the Targeted Review of Internal Models (TRIM), and others, along with best practices in the highly regulated area of credit risk applications. While our recommendations do not focus on the distinction between uses of ML in the private versus public sectors, or the differing needs that each sector may require, others have considered the differences in these contexts. For a discussion regarding governance in public agencies, see Alex Campolo, Madelyn Sanfilipo, Meredith Whittaker, and Kate Crawford, eds. Andrew Selbst and Solon Barocas, "The 2017 AI Now Report," available at https://ainowinstitute.org/AI_Now_2017_Report.pdf, and Dillon Reisman, Jason Schultz, Kate Crawford, and Meredith Whittaker, "Algorithmic Impact Assessments: A Practical Framework for Public Agency Accountability," April 2018, available at https://ainowinstitute.org/aiareport2018.pdf. Similarly, our recommendations do not distinguish between contracted support (for smaller organizations) and in-house teams (in larger organizations), and how such differences almost certainly affect the scale and scope of the practices we suggest.

[4] Note that technical solutions to the "black box" problem of understanding machine learning models tend to focus on either post-hoc reverse engineering of explanations or transparent model design; less frequently do they focus on this foundational trade off. For a survey of such methods, see Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti, "A Survey Of Methods For Explaining Black Box Models," February 2018, available at https://arxiv.org/pdf/1802.01933.pdf. For a review of why explainability is an intuitive, but insufficient, response to the problems of "black box" algorithms, see also, Andrew Selbst and Solon Barocas, "The Intuitive Appeal of Explainable Machines," Fordham Law Review (forthcoming), available at https://papers.ssrn.com/sol3/Papers.cfm?abstract_id=3126971.

[5] When it comes to the tradeoff between accuracy and explainability, we recommend, at minimum, documenting how and why a particular model was chosen, along with data supporting this decision, including a legal and technical analysis. Other relevant information about model choice should be documented as well, including the mathematical specifications, techniques, and approximations used by the underlying model. Especially important are any known or potential limitations involved in these techniques.

[6] In "Hidden Technical Debt in Machine Learning Systems," a paper we recommend to ML practitioners, a group of ML researchers at Google explain the problem as such:

> Oftentimes, a prediction from a machine learning model m_a is made widely accessible, either at runtime or by writing to files or

> logs that may later be consumed by other systems. Without access controls, some of these consumers may be undeclared, silently using the output of a given model as an input to another system. In more classical software engineering, these issues are referred to as visibility debt.
>
> Undeclared consumers are expensive at best and dangerous at worst, because they create a hidden tight coupling of model m_a to other parts of the stack. Changes to m_a will very likely impact these other parts, potentially in ways that are unintended, poorly understood, and detrimental . . . . Undeclared consumers may be difficult to detect unless the system is specifically designed to guard against this case.

See D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Françios Crespo, and Dan Dennison, "Hidden Technical Debt in Machine Learning Systems," in Neural Information Processing Systems (NIPS) 2015, available at https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf.

[7] Under SR 11-7, for example, model validation is one such separate but crucial step. According to SR 11-7:

> Validation involves a degree of independence from model development and use. Generally, validation should be done by people who are not responsible for development or use and do not have a stake in whether a model is determined to be valid. . . . As a practical matter, some validation work may be most effectively done by model developers and users; it is essential, however, that such validation work be subject to critical review by an independent party, who should conduct additional activities to ensure proper validation. Overall, the quality of the process is judged by the manner in which models are subject to critical review. This could be determined by evaluating the extent and clarity of documentation, the issues identified by objective parties, and the actions taken by management to address model issues.

See Supervisory Guidance on Model Risk Management, Board of Governors of the Federal Reserve System & Office of the Comptroller of the Currency, April 2011, available at https://www.federalreserve.gov/supervisionreg/srletters/sr1107a1.pdf.

Other model risk management frameworks this paper has drawn from extensively include: Directive 2013/36/EU of the European Parliament and of the Council of 26 June 2013 on access to the activity of credit institutions and the prudential supervision of credit institutions and investment firms; Regulation No. 575/2013 of the European Parliament and of the Council of 26 June 2013 on prudential requirements for credit institutions and investment firms; and the European Central Bank guide to the Targeted Review of Internal Models, often referred to as the "TRIM" guide, and cited in an earlier footnote, among others.

Note that the "three lines of defense" is also a commonly used framework in the world of risk management in auditing. See, for example, The Three Lines of Defense In Effective Risk Management and Control, Institute of Internal Auditors Position Paper, January 2013, available at https://na.theiia.org/standards-guidance/Public%20Documents/PP%20The%20Three%20Lines%20of%20Defense%20in%20Effective%20Risk%20Management%20and%20Control.pdf.

[8] Model governance committees typically form a key component of internal model risk management strategies. As far as we can tell, however, there is no clear standard for how such committees should function in practice. A good case study involves Sifibank, as detailed in Clifford Rossi's A Risk Professional's Survival Guide: Applied Best Practices in Risk Management (2014).

[9] In cases where models make especially important or impactful decisions, such as in the world of medicine, organizations should also consider how individuals subject to those decisions might be able to object or to receive a baseline explanation for such decisions and their consequences. This will form a critical factor for organizations applying ML in the European Union, as the mandates of the EU's General Data Protection Regulation – as set forth in Articles 13-15 and 22 of that regulation, in particular – govern how individuals should meaningfully interact with models that make impactful decisions with their data.

Note, also, that in regards to maturity of a model, just because a model has been deployed historically in connection with a particular use case does not make it automatically low risk. These models also deserve close re-examination.

[10] A key challenge in implementing lines of defense successfully lies in incentivizing members of each line to focus on subject matter beyond their immediate expertise. That is, in practice, personnel on different teams may not be fully informed about specific technical matters, or may not fully respond to the technical recommendations made by other members with different specializations. The best way to protect from this type of "expertise territorialism" is to comingle various job functions throughout each line of defense.

With regards to specific compliance burdens, we note that different sectors and geographic locations have varying compliance burdens associated with models, with the finance and medical sectors being perhaps the most heavily regulated, along with regions like

the European Union. For an in-depth review of explainability under EU's major data regulation, the General Data Protection Regulation, see Bryan Casey, Ashkon Farhangi, and Roland Vogl, "Rethinking Explainable Machines: The GDPR's 'Right to Explanation' Debate and the Rise of Algorithmic Audits in Enterprise," Berkeley Technology Law Journal (forthcoming), available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3143325.

[11] This includes managing the authorizations and permissioning required to implement proper access controls.

[12] Creating useful model documentation itself can be quite challenging, as this requires keeping track of parameters, hyperparameters, and other configurations of the model. There may also be a large number of trained models that were considered and ultimately discarded for a slightly different version with better performance. There are some nascent solutions to this problem. For example, Predictive Model Markup Language, or PMML, creates a way to semantically label different parts of a model. The larger goal is, ultimately, to make models portable between platforms using an XML file, rather than requiring engineers to wrangle the model output from one platform to another, without the loss of important model metadata.

[13] Patrick Hall, Wen Phan, and SriSatish Ambat provide a good summary of these techniques, and others, in "Ideas on Interpreting Machine Learning," O'Reilly Media, March 15, 2017, available at https://www.oreilly.com/ideas/ideas-on-interpreting-machine-learning. Leave-one-feature-out evaluations of the model can also be used to highlight significant input features that might lead to bias, which we describe in further detail in the following section.

[14] Changes in model behavior based on changes in data are not always undesirable. In fact, for ML systems, change may reflect positive model evolution. That said, understanding that certain changes have occurred and making a judgment on whether the change is desirable, undesirable, or unclear is crucial.

[15] Output analysis may also include evaluation for detection compensation, subverted self-learning, and evidence of workarounds during development instead of fixes. In addition, documentation should include what standards were used for setting threshold parameters and the justification for those standards. In particular, data scientists should document their process for setting acceptable false positive and false negative rates, including any trials run using alternate versions of the model. This documentation should continue as the model evolves and as new risks potentially come to light.

[16] This type of feature proxy analysis should be tailored to the initial objectives—in particular, the initial undesired outcomes—set forth by legal personnel. Only by knowing exactly what to avoid, or what outcomes to minimize, can data scientists assess and test their models appropriately. We have included these particular recommendations in this whitepaper's section on output data, though we are well aware that they may just as well belong in the above section. Note, also, that financial institutions have traditionally relied on disparate impact analysis, which evaluates the adverse impact of facially neutral practices on certain protected classes when assessing bias and discrimination.

[17] Martin Zinkevich provides a good overview of what he calls training-serving skew, defined as "the difference between performance during training and performance during serving," which such monitoring can help detect. We highly recommend his overview of best practices for implementing ML in Martin Zinkevich, "Rules of Machine Learning: Best Practices for ML Engineering," available at http://martin.zinkevich.org/rules_of_ml/rules_of_ml.pdf.

[18] Note that Zinkevich specifically advocates building models slowly, and beginning with hardcoded business logic to address this issue. By doing so, model or system changes can be better understood over time. If models are well-documented and stored, this type of incremental change also more easily facilitates rolling back a production model not performing as expected or desired.

[19] In cases where a model is pulled from production, the reasons for pulling the model and the processes for addressing or fixing any errors in the model should also be documented in detail. We recommend that organizations have a process in place to provide notice to consumers of a model once pulled from production. When a model or dataset is pulled from production, or is replaced by another model or dataset, end users of the system should be notified promptly with the reason the model is being pulled, an explanation of the alternative model and its potential impact on performance, and the steps the team is taking to address the issue.