Responsible Use of Machine Learning

At AWS, we are proud to support our customers as they invent, build, and use machine learning (ML) systems to solve real-world problems. We see the transformational nature of ML technology across industries every day. ML techniques can make tasks easier, safer, and more efficient. For example, ML has been used to develop transcription and translation services, fraud detection software, search and recommendation engines, and tools that monitor and help protect our environment.

Given the breadth and depth of ML, many customers are asking for perspectives on how to responsibly develop and use ML systems. This document shares some recommendations, examples, and tools that can be used across three major phases of ML life cycles: (1) design and development; (2) deployment; and (3) ongoing use.

An important preliminary note is that we believe all use of ML must respect the rule of law, human rights, and values of equity, privacy, and fairness. The field of responsible ML is a rapidly developing area, so these recommendations should be viewed as a starting point and not the final answer. We encourage readers to consider the spirit and intent behind the recommendations. They should be considered along with third party and AWS tools and resources on responsible development and use of ML systems, such as the ones listed below. We are also eager to receive feedback, and appreciate the opportunity to contribute to this important topic while continuing to learn from the broader community.

Phase 1: Design and Development

This phase includes establishing requirements of the ML system, defining performance criteria, exploring the potential impact of the system on users and other parties, collecting and curating training data, and building and testing models and other system components.

Evaluating Use Cases: There are a wide variety of use cases that may incorporate ML, with different goals, characteristics, user bases, and potential impacts. Developers should consider the benefits and potential risks of their specific use case. Given the broad nature and applicability of ML, many applications may pose limited or no risk (e.g., movie recommendation systems), while others could involve significant risk, especially if used in a way that impacts human rights or safety. Examples of risks worth carefully evaluating include: technical limitations of an ML system, over-reliance on limited data or inaccurate output, the potential for bias in training data or the model itself, and intentional or unintentional misuse, as well as the likelihood and impact of those risks and possible solutions. Potential mitigation options may include detailed documentation, explicit warnings or contractual restrictions, technical restrictions, or mechanisms to receive and act on feedback.

ML Capabilities and Limitations: Developers and users should understand the nature, capabilities, and limitations of ML systems, including important concepts like the probabilistic nature of ML, confidence levels, and human review. Many ML systems predict a possible or likely answer, not the answer itself. The probabilistic nature of ML means that use cases that require definitive answers (as opposed to possible or likely answers) may benefit from additional guardrails. Consider providing a numeric or other indicator of the confidence level associated with system output to help users evaluate the output for their use case. Also consider whether human review or oversight of the system may be appropriate, and when it should be required. As an example, if an ML system helps predict the risk of fraud in online transactions, it may not be appropriate to take output from the system as the sole indicator of fraud, but as one factor to be analyzed in connection with the overall transaction. In certain cases it may be appropriate for a trained person to review the ML prediction and transaction before any action is taken. In cases where human review is needed, consider how to provide reviewers the necessary training, context, and interface to take action.

Building and Training Diverse Teams: It is important to have diverse backgrounds, perspectives, skills, and experiences on teams that are developing ML systems. Assess whether teams include a wide array of genders, races, ethnicities, abilities, ages, religions, sexual orientations, military status, backgrounds, and political views. Further assess whether teams may have gaps and consider adding underrepresented perspectives to fill those gaps to enhance performance. Successful teams will likely have cross-functional expertise (e.g., technologists, academics, industry experts, lawyers, and other stakeholders) and diverse characteristics to help ensure important perspectives are taken into account. Consider resources such as user testing, focus groups, and/or third party advocacy groups to obtain additional perspectives from outside parties. There are many public resources, such as the **EU Assessment List for Trustworthy Artificial Intelligence**, that can enable deeper analyses on these subjects.

Be Mindful of Overall Impact: Consider the potential impact of an ML system on parties that are not customers or direct users of the system, but may still be affected. For example, if an autonomous vehicle is not operating as expected, it could have an impact on passengers, other drivers, pedestrians, or property. Similarly, consider guidelines or restrictions for use of ML systems that determine whether users are eligible for certain services or benefits, if those users may lose eligibility based on how the ML system's output is used.

Data Collection: Consider how you will acquire data to develop and test ML models. For example, data may be available through open source repositories, through licenses from third party data providers, or already in your possession. Involve your legal and procurement teams as appropriate to assess the impact of any privacy considerations or other relevant laws, license, or contractual requirements that may impact your collection or use of the data. Consider any necessary processes for handling data securely and safely, and ways to mitigate risk. For example, if certain portions of a data set are sensitive, but are not necessary for development of the model, consider whether you can discard that content.

Training and Testing Data: When collecting and evaluating data to develop and test models, consider its completeness, representativeness, and breadth. Diversity of data is often important for use cases that involve personal characteristics like race and gender, but can also apply in non-obvious contexts. Develop mechanisms to evaluate whether the data appropriately represents real world use, and collect and test additional data to address underrepresented attributes. For example, an audio transcription system may need data with different accents, speech speeds, vernacular, and background environments, and autonomous transport systems may need data from different terrains and obstacles (e.g., cobblestones, dirt, and cracked sidewalks). Review data for freshness (data may be outdated and in need of replacement), potential sources of error (inherent to the data itself, in its structure and organization, or introduced during annotation), and bias (discussed below). It is important to have separate sets of data for training and testing, but both sets should be complete and representative.

Bias: Consider ways to maximize accuracy and reduce bias in data, algorithms, and system design. Some suggestions include:

- Staffing development and annotation teams with a diverse set of backgrounds, perspectives, skills, experiences, and demographics as appropriate for your system's use case and performance.
- Understanding perspectives and potential biases of data annotators and developers, and having processes to mitigate human error (for example, by using annotators familiar with the subject matter, being thoughtful when using labels that require subjective versus objective judgment, and checking annotation samples for accuracy). Consider using multiple data annotators and developers to help identify discrepancies.
- Creating fairness goals and metrics (including potential minimum acceptable thresholds) to measure performance across different subgroups, communities, and demographics applicable to the use case, and testing and measuring progress against those metrics. For example, a speech recognition system may be evaluated for accuracy across different speaker groups by running statistical studies to determine the correlation between speaker demographic variables (such as regional accents) and error rates. Consider whether and how bias is measured in existing processes that do not use ML, and how to use that information to evaluate the effectiveness of the ML system.
- Having independent teams help test the system for bias, and considering whether it may be appropriate or feasible to have external parties perform such evaluations.
- Developing plans to remediate potential inaccuracies and bias, which may include evaluating root causes, developing new requirements, acquiring more data, and re-training models.
- Considering mechanisms to allow users to evaluate system performance and bias/accuracy using their own data for their specific use case.

Explainability of ML systems: Consider the need to explain the methodology and important factors that influence the ML system's output. Mechanisms to help explain complex ML models are still being researched and there is currently no "silver bullet" for explainability, but some areas (like explainability of models that use structured tabular data) have progressed further than others and can be used to help explain certain predictions today. The importance of explainability will vary depending on the use case: many systems that have low or no risk may not require explainability, while ML systems whose output may be used in a manner that could impact human rights or safety will likely need a method for determining how the system performed its analysis. If explainability is not technically feasible, consider whether other mechanisms, such as human review, auditability (next section), and re-focusing or limiting the scope of the use case might serve as an appropriate alternative.

Auditability: Consider the need for implementing mechanisms to track and review steps taken during development and operation of the ML system, e.g., to trace root causes for problems or meet governance requirements. Evaluate the need to document relevant design decisions and inputs to assist in such reviews. Establishing a traceable record can help internal or external teams evaluate the development and functioning of the ML system.

Legal Compliance: Engage with legal advisors to assess requirements for and implications of building your ML system. This may include vetting legal rights to use data and models, and determining applicability of laws around privacy, biometrics, anti-discrimination, and other use-case specific regulations. Be mindful of differing legal requirements across states, provinces, and countries, as well as new AI/ML regulation being considered and proposed around the world. Re-visit legal requirements and considerations through future deployment and operations phases.

Phase 2: Deployment

This phase includes preparing and deploying ML systems for use, including understanding and accounting for capabilities, limitations, and risks associated with deployment.

Education, Documentation and Training: Consider whether users and other stakeholders should be educated on topics like the predictive nature of ML, confidence indicators and thresholds, capabilities and limitations of the system, recommended or prohibited uses, and best practices. As an example, if deploying a conversational chatbot, users should be informed that they are interacting with a computer system and not a real person, to avoid misunderstandings about the nature of the interaction. If using a facial recognition system to assist personnel in making decisions that could impact a person's civil liberties or human rights, include appropriate training for human reviewers on the nature and proper use of such systems. Consider appropriate mechanisms and processes for carrying out trainings or communications, and the need to provide more educational resources around issues like privacy, safety, transparency, accessibility, inclusiveness, and bias.

Confidence Levels and Human Review: As noted earlier, it's important to understand that many ML systems generate predictions of a possible or likely answer, not the answer itself. If confidence indicators are available, take them into account (or instruct your users to take them into account) when reviewing and taking action using output provided by the system. For higher risk use cases, be mindful of situations where confidence indicators may not be appropriately considered and may instead be used as a shortcut to make decisions, and whether such behavior can be mitigated. Regardless of confidence levels, consider whether human review or oversight over the operation of the system may be appropriate or necessary (e.g., in situations where ML systems may be used in a manner that impact human rights or safety), and if so, how to best incorporate such human input into the overall operation of the system. Human reviewers should be appropriately trained on real world scenarios, including examples where the system fails to properly process inputs or cannot handle edge cases, and have ways to exercise meaningful oversight.

Use Case Evaluation and Testing: Consider whether a particular ML model is appropriate for the use case, including any benefits, limitations, and risks. This should be reassessed if the model is used for new use cases, or beyond the system for which it was designed. It is important to test ML systems in the operational environments and on the data on which they will be deployed before live deployment. Develop metrics and a test plan to measure performance of the system against production uses, and consider ongoing tests against a frequently updated "gold standard" dataset. Testing should include not just the ML system itself but also the overall process it is a part of, including decisions or actions that might be taken based on system output. In some situations, it may not be appropriate to use the system if testing does not reach a specified accuracy level. In other cases, such as where the system is used for entertainment purposes or to "narrow the field" for additional review or human judgment, accuracy is one variable that should be balanced with other factors, such as the need to generate a large number of results. Deployers should also factor in localization requirements when deploying an ML system into a new use case, region, or geography different from the one for which it was designed and tested -- for example, real estate pricing models in different geographic areas, or voice recognition systems deployed in areas with different dialects or accents.

Notice and Accessibility: Consider whether to notify end users about the use of ML in the system they are interacting with, such as the earlier example about notifying users that they are interacting with a chatbot and not a live human. Consider whether it is appropriate or feasible to allow end users to bypass interacting with the system and offer an alternate method to accomplish the use case -- for example, some users may prefer not to use a facial recognition authentication system and request a different method of authentication. Consult accessibility resources to ensure that the system is actually usable by the target audience and provides appropriate access options to all intended users.

Operational Data: Consider the sources of any data used with the ML model once it is deployed. As with data used for training and testing, involve your legal and procurement teams as appropriate to assess the impact of any relevant laws or contractual requirements on operational data. Consider any necessary processes for handling data securely and safely, and ways to mitigate risk.

Safety, Security, and Robustness: Use of the ML system must be safe for both users and third parties. As with any technology, deployers should implement appropriate mechanisms to protect the ML system and associated data (both inputs and outputs) from loss, attack, vulnerabilities, or unexpected or malicious user behavior. These mechanisms may include limiting potential access to the system, putting in place legal or technical restrictions on use, and/or implementing warnings, notices, education and trainings that educate users about risks and consequences of improper use. Consider how potential inaccuracies in results produced by the ML system may impact users and relevant stakeholders, and prepare a plan for addressing these inaccuracies, which may involve narrowing the scope of use, relying on human review or oversight, or altering dependencies on the system.

Legal Compliance: As noted in the development phase, it is important to engage your legal advisors to assess legal requirements arising from your deployment and use of the system.

Phase 3: Operation

This phase deals with ongoing operation of the system after it is developed and deployed. Note that many considerations and questions from Phases 1 and 2 continue to be relevant.

Provide and Use Feedback Mechanisms: Since ML systems can continue to “learn” and improve throughout their lifecycle, an important aspect of improvement involves receiving and incorporating feedback from users and stakeholders. Consider soliciting feedback through programmatic and manual methods, including in-system mechanisms or third party outreach through surveys and focus groups. Keep in mind that not all feedback will be relevant or actionable, and it may be appropriate to develop and communicate expectations for acknowledging and addressing feedback. If appropriate for the use case (such as if an ML system might be used to help make decisions on eligibility for important services), consider mechanisms for users or stakeholders to request more information about, or obtain remediation for, negative impact arising from how system output is used.

Continuous Improvement and Validation: ML is an iterative science. Consider the issues raised in previous phases about monitoring and testing of your system. ML models can be subject to “concept drift,” where model behavior changes as a result of changes in users, environments, or data over time. There are multiple ways that models in ML systems can drift, including changes to the use case, operating environment, or types and quality of data. Develop and run ongoing performance tests, and use these test results and feedback to identify areas where additional data or development may improve your system’s performance. Be thoughtful about the data being used as inputs and for any further training or tuning of the ML system. Continue to monitor for potential bias and accuracy, including that your models perform as expected across different segments. Consider appropriate adjustments to both the system and overall processes that involve the system, such as updated training, new notices or restrictions, or optimizing the ways system output is evaluated and used.

Ongoing Education: ML is a constantly evolving landscape, and new techniques, technologies, laws, and social norms will continue to be developed and refined over time. It is critical that all parties involved with building and using ML systems stay educated on these issues and account for them in the design, deployment, and operation of their systems. We encourage all stakeholders in the field, and other interested parties, to contribute knowledge and relay their experiences and learnings to the broader community.

Tools and Resources

AWS offers a large number of tools and resources to help you responsibly develop and use ML systems, including methods to help address some of the issues above.

Amazon SageMaker Clarify provides ML developers with greater visibility into their training data and models so they can identify and limit bias and explain predictions. Amazon SageMaker Clarify detects potential bias during data preparation, after model training, and in a deployed model by examining attributes you specify. For instance, you can check for bias related to age in your initial dataset or in your trained model and receive a detailed report that quantifies different types of possible bias. SageMaker Clarify also helps you to look at the importance of model inputs to explain why models make the predictions they do. SageMaker Clarify includes feature importance graphs that help you explain model predictions and produces reports which can be used to support internal presentations or to identify issues with your model that you can take steps to correct.

Amazon Augmented AI makes it easy to build the workflows required for human review of ML systems. Amazon A2I brings human review to all developers, removing the difficult tasks of building custom human review systems or managing large numbers of human reviewers. As mentioned above, in some situations it may be appropriate to have human oversight over ML systems to help ensure accuracy, provide continuous improvements, or retrain models with updated predictions. Amazon A2I streamlines building and managing human reviews for ML applications. Amazon A2I provides built-in human review workflows for common ML use cases, such as content moderation and text extraction from documents. You can also create your own workflows for ML models built on SageMaker or any other tools. Using Amazon A2I, you can allow human reviewers to step in when a model is unable to make a high-confidence prediction or to audit its predictions on an ongoing basis.

Amazon SageMaker Model Monitor helps maintain high quality ML models by detecting model and concept drift in real-time, and sending alerts to enable immediate action. Model and concept drift are detected by monitoring the quality of the model based on independent and dependent variables. Independent variables (also known as features) are the inputs to an ML model, and dependent variables are the outputs of the model. For example, with an ML model predicting a bank loan approval, independent variables could be age, income, and credit history of the applicant, and the dependent variable would be the actual result of the loan application. SageMaker Model Monitor constantly monitors model performance characteristics such as accuracy, which measures the number of correct predictions compared to the total number of predictions, so you can take action to address anomalies.

Amazon SageMaker Data Wrangler gives you better control of your training and testing data by simplifying the process of data preparation and feature engineering. You can complete each step of the data preparation workflow, including data selection, cleansing, exploration, and visualization from a single visual interface. It also helps you identify potential errors, extreme values, and inconsistencies in your data preparation workflow through visualization templates. Using SageMaker Data Wrangler's data selection tool, you can choose the data you want from various data sources and import it with a single click. SageMaker Data Wrangler contains over 300 built-in data transformations so you can quickly normalize, transform, and combine features without having to write any code.

Training and Professional Services. AWS offers the latest in ML education through the Machine Learning University, Training and Certification program, and AWS ML Embark. You can also work with experts in responsible ML within our AWS Professional Services organization to create an operational approach encompassing people, processes, and technology to develop and operationalize responsible ML principles based on a proven framework. This can helps you look around corners, uncover potential unintended impacts, and mitigate risks related to the development, deployment, and operationalization of ML systems.

Research, Innovation, and External Collaboration. AWS collaborates with academia and other stakeholders through strategic partnerships with universities including University of California, Berkeley, MIT, California Institute of Technology, the University of Washington, and others. We are also active members of multi-stakeholder organizations relating to AI, including OECD AI working groups and The Partnership on AI. We also provide research grants through Amazon Research Awards and the joint Amazon and National Science Foundation Fairness in AI Grants program. See some of our recent videos and publications related to Responsible AI:

Amazon scientist Dr. Nashlie Sephus focuses on ensuring accuracy in machine learning

Amazon Scholars Michael Kearns and Aaron Roth discuss the ethics of machine learning

How a paper by three Oxford academics influenced AWS bias and explainability software

Nine videos about explainable AI in industry

Related Publications From Amazon Science:

Correcting exposure bias for link recommendation

Fair Bayesian optimization

General Fair Empirical Risk Minimization

Learning Deep Fair Graph Neural Networks

Bias preservation in machine learning: The legality of fairness metrics under EU non-discrimination law

Learning fair and transferable representations with theoretical guarantees

Exploiting MMD and Sinkhorn divergences for fair and transferable representation learning

Towards unbiased and accurate deferral to multiple experts

Amazon SageMaker Clarify: Machine learning bias detection and explainability in the cloud

Learning to rank in the position based model with bandit feedback

Decoding and diversity in machine translation

Fairness measures for machine learning in finance

Fair Bayesian optimization

Mixed-privacy forgetting in deep networks

Continuous compliance