RESPONSIBILITIES ›

Responsible AI Practices

The development of AI is creating new opportunities to improve the lives of people around the world, from business to healthcare to education. It is also raising new questions about the best way to build fairness, interpretability, privacy, and security into these systems.

These questions are far from solved, and in fact are active areas of research and development. Google is committed to making progress in the responsible development of AI and to sharing knowledge, research, tools, datasets, and other resources with the larger community. Below we share some of our current work and recommended practices. As with all of our research, we will take our latest findings into account, work to incorporate them as appropriate, and adapt as we learn more over time.

AI and advanced technologies at Google

Explore our commitment to the responsible development and use of AI.

General recommended practices for AI

Reliable, effective user-centered AI systems should be designed following general best practices for software systems, together with practices that address considerations unique to machine learning. Our top recommendations are outlined below, with additional resources for further reading.

Recommended practices

Use a human-centered design approach

The way actual users experience your system is essential to assessing the true impact of its predictions, recommendations, and decisions.

- Design features with appropriate disclosures built-in: clarity and control is crucial to a good user experience.
- Consider augmentation and assistance: producing a single answer can be appropriate where there is a high probability that the answer satisfies a diversity of users and use cases. In other cases, it may be optimal for your system to suggest a few options to the user. Technically, it is much more difficult to achieve good precision at one answer (P@1) versus precision at a few answers (e.g., P@3).
- Model potential adverse feedback early in the design process, followed by specific live testing and iteration for a small fraction of traffic before full deployment.
- Engage with a diverse set of users and use-case scenarios, and incorporate feedback before and throughout project development. This will build a rich variety of user perspectives into the project and increase the number of people who benefit from the technology.

Identify multiple metrics to assess training and monitoring

The use of several metrics rather than a single one will help you to understand tradeoffs between different kinds of errors and experiences.

- Consider metrics including feedback from user surveys, quantities that track overall system performance and short- and long-term product heath (e.g., click-through rate and customer lifetime value, respectively), and false positive and false negative rates sliced across different subgroups.
- Ensure that your metrics are appropriate for the context and goals of your system, e.g., a fire alarm system should have high recall, even if that means the occasional false alarm.

When possible, directly examine your raw data

ML models will reflect the data they are trained on, so analyze your raw data carefully to ensure you understand it. In cases where this is not possible, e.g., with sensitive raw data, understand your input data as much as possible while respecting privacy; for example by computing aggregate, anonymized summaries.

- Does your data contain any mistakes (e.g., missing values, incorrect labels)?
- Is your data sampled in a way that represents your users (e.g., will be used for all ages, but you only have training data from senior citizens) and the real-world setting (e.g., will be used year-round, but you only have training data from the summer)? Is the data accurate?
- Training-serving skew—the difference between performance during training and performance during serving—is a persistent challenge. During training, try to identify potential skews and work to address them, including by adjusting your training data or objective function. During evaluation, continue to try to get evaluation data that is as representative as possible of the deployed setting.
- Are any features in your model redundant or unnecessary? Use the simplest model that meets your performance goals.
- For supervised systems, consider the relationship between the data labels you have, and the items you are trying to predict. If you are using a data label X as a proxy to predict a label Y, in which cases is the gap between X and Y problematic?
- Data bias is another important consideration; learn more in practices on AI and fairness.

Understand the limitations of your dataset and model

- A model trained to detect correlations should not be used to make causal inferences, or imply that it can. E.g., your model may learn that people who buy basketball shoes are taller on average, but this does not mean that a user who buys basketball shoes will become taller as a result.
- Machine learning models today are largely a reflection of the patterns of their training data. It is therefore important to communicate the scope and coverage of the training, hence clarifying the capability and limitations of the models. E.g., a shoe detector trained with stock photos can work best with stock photos but has limited capability when tested with user-generated cellphone photos.
- Communicate limitations to users where possible. For example, an app that uses ML to recognize specific bird species might communicate that the model was trained on a small set of images from a specific region of the world. By better educating the user, you may also improve the feedback provided from users about your feature or application.

Test, Test, Test

Learn from software engineering best test practices and quality engineering to make sure the AI system is working as intended and can be trusted.

- Conduct rigorous unit tests to test each component of the system in isolation.
- Conduct integration tests to understand how individual ML components interact with other parts of the overall system.
- Proactively detect input drift by testing the statistics of the inputs to the AI system to make sure they are not changing in unexpected ways.

not changing in unexpected ways.

- Use a gold standard dataset to test the system and ensure that it continues to behave as expected. Update this test set regularly in line with changing users and use cases, and to reduce the likelihood of training on the test set.
- Conduct iterative user testing to incorporate a diverse set of users' needs in the development cycles.
- Apply the quality engineering principle of poka-yoke: build quality checks into a system, so that unintended failures either cannot happen or trigger an immediate response (e.g., if an important feature is unexpectedly missing, the AI system won't output a prediction).

Continue to monitor and update the system after deployment

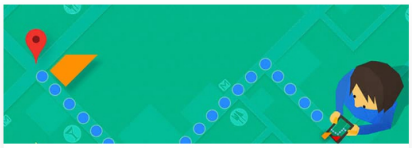
Continued monitoring will ensure your model takes real-world performance and user feedback (e.g., happiness tracking surveys, HEART framework) into account.

- Issues will occur: any model of the world is imperfect almost by definition. Build time into your product roadmap to allow you to address issues.
- Consider both short- and long-term solutions to issues. A simple fix (e.g., blacklisting or whitelisting) may help to solve a problem quickly, but may not be the optimal solution in the long run. Balance short-term simple fixes with longer-term learned solutions.
- Before updating a deployed model, analyze how the candidate and deployed models differ, and how the update will affect the overall system quality and user experience.

Examples of our work

Rules of ML

A practical guide that outlines best practices for ML engineering.

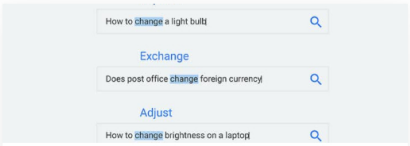


Human-centered ML

Recommended steps to stay focused on the user from our Google UX community.

Machine learning: the high-interest credit card of technical debt

ML-specific risk factors and design patterns to refactor or avoid.



Search quality rating guidelines

Communicates how the Google Search algorithm works and helps webmasters understand quality and ranking.

PAIR

People + AI Guidebook

People + AI Research initiative publishes practical insights on multidisciplinary, human-centered approaches to designing with AI.

Provides best practices for designing human-centered AI products.

Building responsible AI for everyone

Explore our efforts to develop and use AI responsibly in order to benefit people and society.

Learn more