L4_GOV_TargetingLimits

Targeting limits in vulnerable contexts (policy)

Safety settings

The Gemini API provides safety settings that you can adjust during the prototyping stage to determine if your application requires more or less restrictive safety configuration. You can adjust these settings across five filter categories to restrict or allow certain types of content. This guide covers how the Gemini API handles safety settings and filtering and how you can change the safety settings for your application.

Note: Applications that use less restrictive safety settings may be subject to review. See the Terms of Service for more information.

Safety filters

The Gemini API's adjustable safety filters cover the following categories:

Category	Description
Harassment	Negative or harmful comments targeting identity and/or protected attributes.
Hate speech	Content that is rude, disrespectful, or profane.
Sexually explicit	Contains references to sexual acts or other lewd content.
Dangerous	Promotes, facilitates, or encourages harmful acts.
Civic integrity	Election-related queries.

These categories are defined in **HarmCategory**. The Gemini models only support **HARM_CATEGORY_HARASSMENT**, **HARM_CATEGORY_HATE_SPEECH**, **HARM_CATEGORY_SEXUALLY_EXPLICIT**, and **HARM_CATEGORY_DANGEROUS_CONTENT**. All other categories are used only by PaLM 2 (Legacy) models.

You can use these filters to adjust what's appropriate for your use case. For example, if you're building video game dialogue, you may deem it acceptable to allow more content that's rated as *Dangerous* due to the nature of the game.

In addition to the adjustable safety filters, the Gemini API has built-in protections against core harms, such as content that endangers child safety. These types of harm are always blocked and cannot be adjusted.

Content safety filtering level

The Gemini API categorizes the probability level of content being unsafe as HIGH, MEDIUM, LOW, or NEGLIGIBLE.

The Gemini API blocks content based on the probability of content being unsafe and not the severity. This is important to consider because some content can have low probability of

being unsafe even though the severity of harm could still be high. For example, comparing the sentences:

1. The robot punched me.
2. The robot slashed me up.

The first sentence might result in a higher probability of being unsafe, but you might consider the second sentence to be a higher severity in terms of violence. Given this, it is important that you carefully test and consider what the appropriate level of blocking is needed to support your key use cases while minimizing harm to end users.

Safety filtering per request

You can adjust the safety settings for each request you make to the API. When you make a request, the content is analyzed and assigned a safety rating. The safety rating includes the category and the probability of the harm classification. For example, if the content was blocked due to the harassment category having a high probability, the safety rating returned would have category equal to HARASSMENT and harm probability set to HIGH.

By default, safety settings block content (including prompts) with medium or higher probability of being unsafe across any filter. This baseline safety is designed to work for most use cases, so you should only adjust your safety settings if it's consistently required for your application.

The following table describes the block settings you can adjust for each category. For example, if you set the block setting to **Block few** for the **Hate speech** category, everything that has a high probability of being hate speech content is blocked. But anything with a lower probability is allowed.

Threshold (Google AI Studio)	Threshold (API)	Description
Off	`OFF`	Turn off the safety filter
Block none	`BLOCK_NONE`	Always show regardless of probability of unsafe content
Block few	`BLOCK_ONLY_HIGH`	Block when high probability of unsafe content
Block some	`BLOCK_MEDIUM_AND_ABOVE`	Block when medium or high probability of unsafe content
Block most	`BLOCK_LOW_AND_ABOVE`	Block when low, medium or high probability of unsafe content
N/A	`HARM_BLOCK_THRESHOLD_UNSPECIFIED`	Threshold is unspecified, block using default threshold

If the threshold is not set, the default block threshold is **Block none** (for all newer stable GA models) or **Block some** (in all other models) for all categories **except** the *Civic integrity* category.

The default block threshold for the *Civic integrity* category is **Block none** (for gemini-2.0-flash, and gemini-2.0-flash-lite) both for Google AI Studio and the Gemini API, and **Block most** for all other models in Google AI Studio only.

You can set these settings for each request you make to the generative service. See the HarmBlockThreshold API reference for details.

Safety feedback

generateContent returns a GenerateContentResponse which includes safety feedback.

Prompt feedback is included in promptFeedback. If promptFeedback.blockReason is set, then the content of the prompt was blocked.

Response candidate feedback is included in Candidate.finishReason and Candidate.safetyRatings. If response content was blocked and the finishReason was SAFETY, you can inspect safetyRatings for more details. The content that was blocked is not returned.

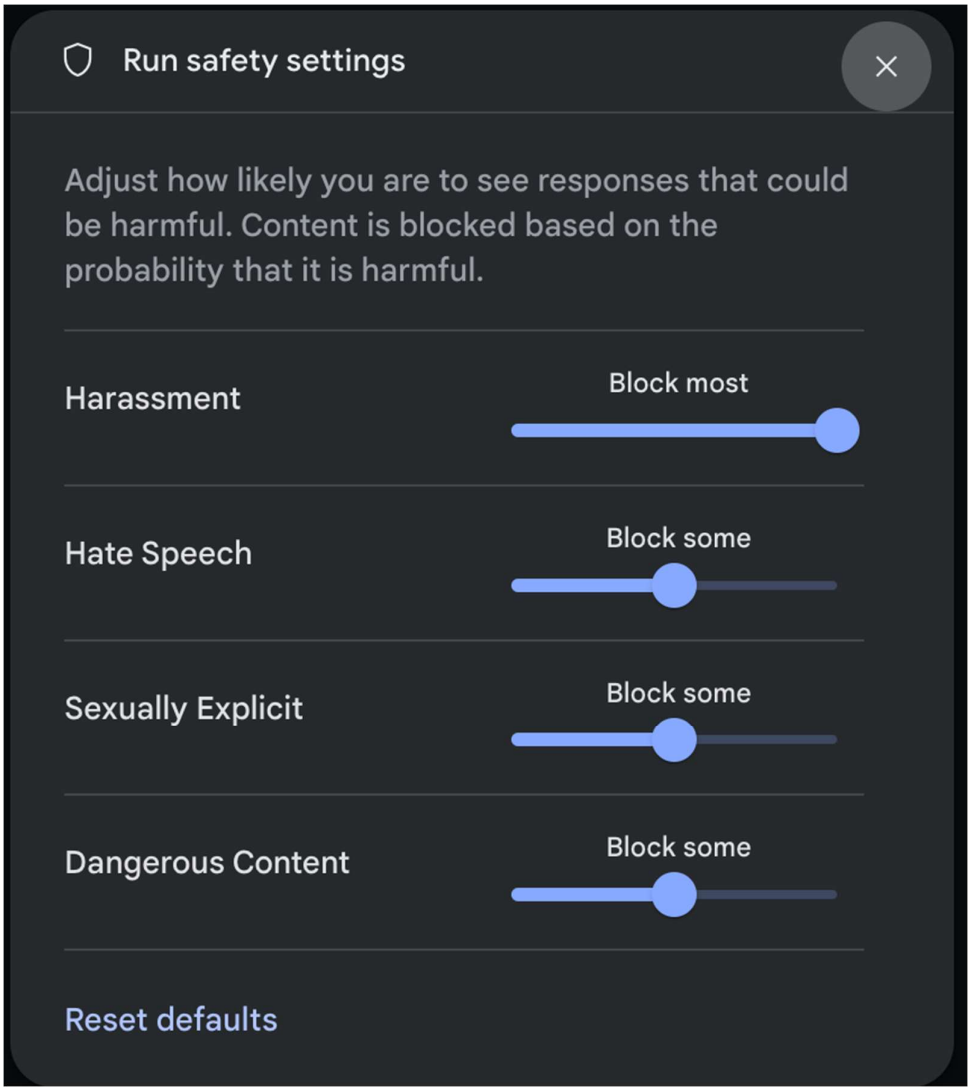
Adjust safety settings

This section covers how to adjust the safety settings in both Google AI Studio and in your code.

Google AI Studio

You can adjust safety settings in Google AI Studio, but you cannot turn them off.

Click **Edit safety settings** in the **Run settings** panel to open the **Run safety settings** modal. In the modal, you can use the sliders to adjust the content filtering level per safety category:



Note: If you set any of the category filters to **Block none**, Google AI Studio will display a reminder about the Gemini API's Terms of Service with respect to safety settings.

When you send a request (for example, by asking the model a question), a warning **No Content** message appears if the request's content is blocked. To see more details, hold the pointer over the **No Content** text and click warning **Safety**.

Gemini API SDKs

The following code snippet shows how to set safety settings in your GenerateContent call. This sets the thresholds for the harassment (HARM_CATEGORY_HARASSMENT) and hate speech (HARM_CATEGORY_HATE_SPEECH) categories. For example, setting these categories to BLOCK_LOW_AND_ABOVE blocks any content that has a low or higher probability of being harassment or hate speech. To understand the threshold settings, see Safety filtering per request.

PythonGoJavaScriptDart (Flutter)KotlinJavaREST

```
from google import genai
from google.genai import types
```

```
import PIL.Image

img = PIL.Image.open("cookies.jpg")

client = genai.Client()

response = client.models.generate_content(
   model="gemini-2.0-flash",
   contents=['Do these look store-bought or homemade?', img],
   config=types.GenerateContentConfig(
    safety_settings=[
     types.SafetySetting(
        category=types.HarmCategory.HARM_CATEGORY_HATE_SPEECH,
        threshold=types.HarmBlockThreshold.BLOCK_LOW_AND_ABOVE,
     ),
    ]
   )
)

print(response.text)
```

Next steps

- See the API reference to learn more about the full API.
- Review the safety guidance for a general look at safety considerations when developing with LLMs.
- Learn more about assessing probability versus severity from the Jigsaw team
- Learn more about the products that contribute to safety solutions like the Perspective API. * You can use these safety settings to create a toxicity classifier. See the classification example to get started.

Additional usage policies

This page includes additional usage policies for the Gemini API.

Abuse monitoring

Google is committed to the responsible development and use of AI. To ensure the safety and integrity of the Gemini API, we have created these policy guidelines. By using the Gemini API, you agree to the following guidelines, the Gemini API Additional Terms of Service and Generative AI Prohibited Use Policy.

How We Monitor for Misuse

Google's Trust and Safety Team employs a combination of automated and manual processes to detect potential misuse of the Gemini API and enforce our policies.

- **Automated Detection:** Automated systems scan API usage for violations of our Prohibited Use Policy, such as hate speech, harassment, sexually explicit content, and dangerous content.
- **Manual Detection:** If a project consistently exhibits suspicious activity, it may be flagged for manual review by authorized Google personnel.

How We Handle Data

To help with abuse monitoring, Google retains the following data for fifty-five (55) days:

- **Prompts:** The text prompts you submit to the API.
- **Contextual Information:** Any additional context you provide with your prompts.
- **Output:** The responses generated by the Gemini API.

How We Investigate Potential Issues

When prompts or model outputs are flagged by safety filters and abuse detection systems described above, authorized Google employees may assess the flagged content, and either confirm or correct the classification or determination based on predefined guidelines and policies. Data can be accessed for human review only by authorized Google employees via an internal governance assessment and review management platform. When data is logged for abuse monitoring, it is used solely for the purpose of policy enforcement and is not used to train or fine-tune any AI/ML models.

Working with You on Policy Compliance

If your use of Gemini doesn't align with our policies, we may take the following steps:

- **Get in touch:** We may reach out to you through email to understand your use case and explore ways to bring your usage into compliance.
- **Temporary usage limits:** We may limit your access to the Gemini API.
- **Temporary suspension:** We may temporarily pause your access to the Gemini API.
- **Account closure:** As a last resort, and for serious violations, we may permanently close your access to the Gemini API and other Google services.

Scope

These policy guidelines apply to the use of the Gemini API and AI Studio.

Inline Preference Voting

In Google AI Studio, you might occasionally see a side-by-side comparison of two different responses to your prompt. This is part of our Inline Preference Voting system. You'll be asked to choose which response you prefer. This helps us understand which model outputs users find most helpful.

Why are we doing this?

We're constantly working to improve our AI models and services. Your feedback through Inline Preference Voting helps us provide, improve, and develop Google products and services and machine learning technologies, including Google's enterprise features, products and services, consistent with the Gemini API Additional Terms of Service and Privacy Policy.

What data is included in Feedback?

To make informed decisions about our models, we collect certain data when you participate in Inline Preference Voting:

- **Prompts and Responses:** We record all prompts and responses, including any uploaded content, in the conversation you submitted feedback about. We also record the two response options that you selected from. This helps us understand the context of your preference.
- **Your Vote:** We record which response you preferred. This is the core of the feedback we're collecting.
- **Usage Details:** This includes information about which model generated the response and other technical and operational details about your usage of this feature.

Your Privacy

We take your privacy seriously. Google takes steps to protect your privacy as part of this process. This includes disconnecting this data from your Google Account, API key, and Cloud project before reviewers see or annotate it. **Do not submit feedback on conversations that include sensitive, confidential, or personal information.**

Generative AI Prohibited Use Policy

Last Modified: December 17, 2024

Generative AI models can help you explore, learn, and create. We expect you to engage with them in a responsible, legal, and safe manner. The following restrictions apply to your interactions with generative AI in the Google products and services that refer to this policy.

1. Do not engage in dangerous or illegal activities, or otherwise violate applicable law or regulations. This includes generating or distributing content that:
    a. Relates to child sexual abuse or exploitation.
    b. Facilitates violent extremism or terrorism.
    c. Facilitates non-consensual intimate imagery.
    d. Facilitates self-harm.
    e. Facilitates illegal activities or violations of law -- for example, providing instructions for synthesizing or accessing illegal or regulated substances, goods, or services.
    f. Violates the rights of others, including privacy and intellectual property rights -- for example, using personal data or biometrics without legally-required consent.
    g. Tracks or monitors people without their consent.
    h. Makes automated decisions that have a material detrimental impact on individual rights without human supervision in high-risk domains -- for example, in employment, healthcare, finance, legal, housing, insurance, or social welfare.

2. Do not compromise the security of others' or Google's services. This includes generating or distributing content that facilitates:
    a. Spam, phishing, or malware.
    b. Abuse of, harm to, interference with, or disruption to Google's or others' infrastructure or services.
    c. Circumvention of abuse protections or safety filters -- for example, manipulating the model to contravene our policies.
3. Do not engage in sexually explicit, violent, hateful, or harmful activities. This includes generating or distributing content that facilitates:
    a. Hatred or hate speech.
    b. Harassment, bullying, intimidation, abuse, or the insulting of others.
    c. Violence or the incitement of violence.
    d. Sexually explicit content -- for example, content created for the purpose of pornography or sexual gratification.
4. Do not engage in misinformation, misrepresentation, or misleading activities. This includes:
    a. Frauds, scams, or other deceptive actions.
    b. Impersonating an individual (living or dead) without explicit disclosure, in order to deceive.
    c. Facilitating misleading claims of expertise or capability in sensitive areas -- for example in health, finance, government services, or the law, in order to deceive.
    d. Facilitating misleading claims related to governmental or democratic processes or harmful health practices, in order to deceive.
    e. Misrepresenting the provenance of generated content by claiming it was created solely by a human, in order to deceive.

We may make exceptions to these policies based on educational, documentary, scientific, or artistic considerations, or where harms are outweighed by substantial benefits to the public.

Policy guidelines for the Gemini app

Our goal for the Gemini app is to be maximally helpful to users, while avoiding outputs that could cause real-world harm or offense. Drawing upon the expertise and processes developed over the years through research, user feedback, and expert consultation on various Google products, we aspire to have Gemini avoid certain types of problematic outputs, such as:

Threats to Child Safety

Gemini should not generate outputs, including Child Sexual Abuse Material, that exploit or sexualize children.

Dangerous Activities

Gemini should not generate outputs that encourage or enable dangerous activities that would cause real-world harm. These include:

- Instructions for suicide and other self-harm activities, including eating disorders.
- Facilitation of activities that might cause real-world harm, such as instructions on how to purchase illegal drugs or guides for building weapons.

Violence and Gore

Gemini should not generate outputs that describe or depict sensational, shocking, or gratuitous violence, whether real or fictional. These include:

- Excessive blood, gore, or injuries.
- Gratuitous violence against animals.

Harmful Factual Inaccuracies

Gemini should not generate factually inaccurate outputs that could cause significant, real-world harm to someone's health, safety, or finances. These include:

- Medical information that conflicts with established scientific or medical consensus or evidence-based medical practices.
- Incorrect information that poses a risk to physical safety, such as erroneous disaster alerts or inaccurate news about ongoing violence.

Harassment, Incitement and Discrimination

Gemini should not generate outputs that incite violence, make malicious attacks, or constitute bullying or threats against individuals or groups. These include:

- Calls to attack, injure, or kill individuals or a group.
- Statements that dehumanize or advocate for the discrimination of individuals or groups based on a legally protected characteristic.
- Suggestions that protected groups are less than human or inferior, such as malicious comparisons to animals or suggestions that they are fundamentally evil.

Sexually Explicit Material

Gemini should not generate outputs that describe or depict explicit or graphic sexual acts or sexual violence, or sexual body parts in an explicit manner. These include:

- Pornography or erotic content.
- Depictions of rape, sexual assault, or sexual abuse.

Of course, context matters. We consider multiple factors when evaluating outputs, including educational, documentary, artistic, or scientific applications.

Making sure that Gemini adheres to these guidelines is tricky: There are limitless ways that users can engage with Gemini, and equally limitless ways Gemini can respond. This is because LLMs are probabilistic, which means they are always producing new and different responses to user inputs. And Gemini's outputs are informed by its training data, which means that Gemini will sometimes reflect the limits of that data. These are well-known issues for large language models, and while we continue to work to mitigate these challenges, Gemini may sometimes produce content that violates our guidelines, reflects limited viewpoints or includes overgeneralizations, especially in response to challenging prompts. We highlight

these limitations for users through a variety of means, encourage users to provide feedback, and offer convenient tools to report content for removal under our policies and applicable laws. And of course we expect users to act responsibly and abide by our prohibited use policy.

As we learn more about how people use the Gemini app and find it most helpful, we will update these guidelines. You can find out more here about our approach to building the Gemini app.

Safety guidance

Generative artificial intelligence models are powerful tools, but they are not without their limitations. Their versatility and applicability can sometimes lead to unexpected outputs, such as outputs that are inaccurate, biased, or offensive. Post-processing, and rigorous manual evaluation are essential to limit the risk of harm from such outputs.

The models provided by the Gemini API can be used for a wide variety of generative AI and natural language processing (NLP) applications. Use of these functions is only available through the Gemini API or the Google AI Studio web app. Your use of Gemini API is also subject to the Generative AI Prohibited Use Policy and the Gemini API terms of service.

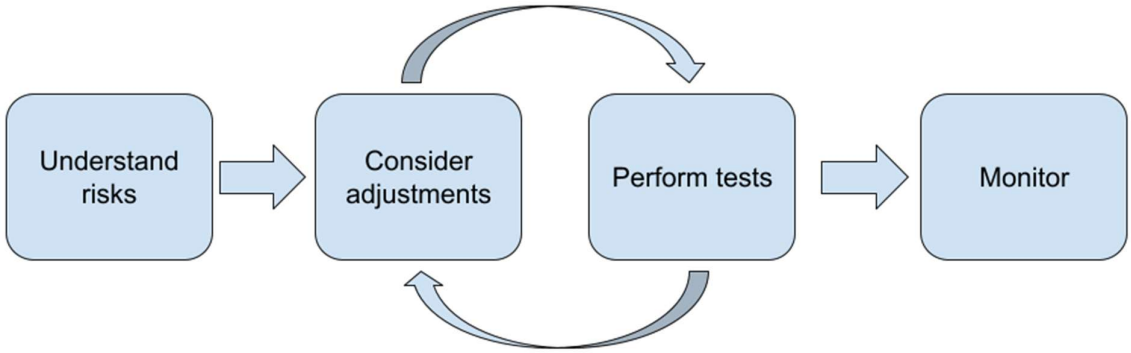
Part of what makes large language models (LLMs) so useful is that they're creative tools that can address many different language tasks. Unfortunately, this also means that large language models can generate output that you don't expect, including text that's offensive, insensitive, or factually incorrect. What's more, the incredible versatility of these models is also what makes it difficult to predict exactly what kinds of undesirable output they might produce. While the Gemini API has been designed with Google's AI principles in mind, the onus is on developers to apply these models responsibly. To aid developers in creating safe, responsible applications, the Gemini API has some built-in content filtering as well as adjustable safety settings across 4 dimensions of harm. Refer to the safety settings guide to learn more.

This document is meant to introduce you to some safety risks that can arise when using LLMs, and recommend emerging safety design and development recommendations. (Note that laws and regulations may also impose restrictions, but such considerations are beyond the scope of this guide.)

The following steps are recommended when building applications with LLMs:

- Understanding the safety risks of your application
- Considering adjustments to mitigate safety risks
- Performing safety testing appropriate to your use case
- Soliciting feedback from users and monitoring usage

The adjustment and testing phases should be iterative until you reach performance appropriate for your application.



Understand the safety risks of your application

In this context, safety is being defined as the ability of an LLM to avoid causing harm to its users, for example, by generating toxic language or content that promotes stereotypes. The models available through the Gemini API have been designed with Google's AI principles in mind and your use of it is subject to the Generative AI Prohibited Use Policy. The API provides built-in safety filters to help address some common language model problems such as toxic language and hate speech, and striving for inclusiveness and avoidance of stereotypes. However, each application can pose a different set of risks to its users. So as the application owner, you are responsible for knowing your users and the potential harms your application may cause, and ensuring that your application uses LLMs safely and responsibly.

As part of this assessment, you should consider the likelihood that harm could occur and determine its seriousness and mitigation steps. For example, an app that generates essays based on factual events would need to be more careful about avoiding misinformation, as compared to an app that generates fictional stories for entertainment. A good way to begin exploring potential safety risks is to research your end users, and others who might be affected by your application's results. This can take many forms including researching state of the art studies in your app domain, observing how people are using similar apps, or running a user study, survey, or conducting informal interviews with potential users.

Advanced tips

Consider adjustments to mitigate safety risks

Now that you have an understanding of the risks, you can decide how to mitigate them. Determining which risks to prioritize and how much you should do to try to prevent them is a critical decision, similar to triaging bugs in a software project. Once you've determined priorities, you can start thinking about the types of mitigations that would be most appropriate. Often simple changes can make a difference and reduce risks.

For example, when designing an application consider:

- **Tuning the model output** to better reflect what is acceptable in your application context. Tuning can make the output of the model more predictable and consistent and therefore can help mitigate certain risks.

- **Providing an input method that facilities safer outputs.** The exact input you give to an LLM can make a difference in the quality of the output. Experimenting with input prompts to find what works most safely in your use-case is well worth the effort, as you can then provide a UX that facilitates it. For example, you could restrict users to choose only from a drop-down list of input prompts, or offer pop-up suggestions with descriptive phrases which you've found perform safely in your application context.
- **Blocking unsafe inputs and filtering output before it is shown to the user.** In simple situations, blocklists can be used to identify and block unsafe words or phrases in prompts or responses, or require human reviewers to manually alter or block such content.
  Note: Automatically blocking based on a static list can have unintended results such as targeting a particular group that commonly uses vocabulary in the blocklist.
- **Using trained classifiers to label each prompt with potential harms or adversarial signals.** Different strategies can then be employed on how to handle the request based on the type of harm detected. For example, If the input is overtly adversarial or abusive in nature, it could be blocked and instead output a pre-scripted response.
  Advanced tip
- **Putting safeguards in place against deliberate misuse** such as assigning each user a unique ID and imposing a limit on the volume of user queries that can be submitted in a given period. Another safeguard is to try and protect against possible prompt injection. Prompt injection, much like SQL injection, is a way for malicious users to design an input prompt that manipulates the output of the model, for example, by sending an input prompt that instructs the model to ignore any previous examples. See the Generative AI Prohibited Use Policy for details about deliberate misuse.
- **Adjusting functionality to something that is inherently lower risk.** Tasks that are narrower in scope (e.g., extracting keywords from passages of text) or that have greater human oversight (e.g., generating short-form content that will be reviewed by a human), often pose a lower risk. So for instance, instead of creating an application to write an email reply from scratch, you might instead limit it to expanding on an outline or suggesting alternative phrasings.
  Perform safety testing appropriate to your use case
  Testing is a key part of building robust and safe applications, but the extent, scope and strategies for testing will vary. For example, a just-for-fun haiku generator is likely to pose less severe risks than, say, an application designed for use by law firms to summarize legal documents and help draft contracts. But the haiku generator may be used by a wider variety of users which means the potential for adversarial attempts or even unintended harmful inputs can be greater. The implementation context also matters. For instance, an application with outputs that are reviewed by human experts prior to any action being taken might be deemed less likely to produce harmful outputs than the identical application without such oversight.

It's not uncommon to go through several iterations of making changes and testing before feeling confident that you're ready to launch, even for applications that are relatively low risk. Two kinds of testing are particularly useful for AI applications:

- **Safety benchmarking** involves designing safety metrics that reflect the ways your application could be unsafe in the context of how it is likely to get used, then testing how well your application performs on the metrics using evaluation datasets. It's good practice to think about the minimum acceptable levels of safety metrics before testing so that 1) you can evaluate the test results against those expectations and 2) you can gather the evaluation dataset based on the tests that evaluate the metrics you care about most.

  Advanced tips

- **Adversarial testing** involves proactively trying to break your application. The goal is to identify points of weakness so that you can take steps to remedy them as appropriate. Adversarial testing can take significant time/effort from evaluators with expertise in your application — but the more you do, the greater your chance of spotting problems, especially those occurring rarely or only after repeated runs of the application.
    - Adversarial testing is a method for systematically evaluating an ML model with the intent of learning how it behaves when provided with malicious or inadvertently harmful input:
        - An input may be malicious when the input is clearly designed to produce an unsafe or harmful output-- for example, asking a text generation model to generate a hateful rant about a particular religion.
        - An input is inadvertently harmful when the input itself may be innocuous, but produces harmful output -- for example, asking a text generation model to describe a person of a particular ethnicity and receiving a racist output.
    - What distinguishes an adversarial test from a standard evaluation is the composition of the data used for testing. For adversarial tests, select test data that is most likely to elicit problematic output from the model. This means probing the model's behaviour for all the types of harms that are possible, including rare or unusual examples and edge-cases that are relevant to safety policies. It should also include diversity in the different dimensions of a sentence such as structure, meaning and length. You can refer to the Google's Responsible AI practices in fairness for more details on what to consider when building a test dataset.

  Advanced tips

- **Note:** LLMs are known to sometimes produce different outputs for the same input prompt. Multiple rounds of testing may be needed to catch more of the problematic outputs.

Monitor for problems

No matter how much you test and mitigate, you can never guarantee perfection, so plan upfront how you'll spot and deal with problems that arise. Common approaches include setting up a monitored channel for users to share feedback (e.g., thumbs up/down rating) and running a user study to proactively solicit feedback from a diverse mix of users — especially valuable if usage patterns are different to expectations.

Links to the papers:

https://ai.google.dev/gemini-api/docs/safety-settings
https://ai.google.dev/gemini-api/docs/usage-policies
https://ai.google.dev/gemini-api/docs/safety-guidance
https://policies.google.com/terms/generative-ai/use-policy
https://gemini.google/policy-guidelines/