**Microsoft (Azure AI Studio)**

**Need a sample dataset in csv. or json. to evaluate the result.**

1. Traditional machine learning metrics
   1. F1 score: measures the ratio of the number of shared words between the model generation and the ground truth answers.
      1. When to use: when you want a single comprehensive metric that combines both recall and precision in your model's responses. It provides a balanced evaluation of your model's performance in terms of capturing accurate information in the response.
2. AI-assisted metrics
   1. Risk and safety metrics: focus on identifying potential content and security risks and ensuring the safety of the generated content. This contains 4 levels of data from very low, low, medium, to high.
      1. Hateful and unfair content defect rate: any language pertaining to hate toward or unfair representations of individuals and social groups along factors including but not limited to race, ethnicity, nationality, gender, sexual orientation, religion, immigration status, ability, personal appearance, and body size. Unfairness occurs when AI systems treat or represent social groups inequitably, creating or contributing to societal inequities.
      2. Sexual content defect rate: language pertaining to anatomical organs and genitals, romantic relationships, acts portrayed in erotic terms, pregnancy, physical sexual acts (including assault or sexual violence), prostitution, pornography, and sexual abuse.
      3. Violent content defect rate: language pertaining to physical actions intended to hurt, injure, damage, or kill someone or something. It also includes descriptions of weapons and guns (and related entities such as manufacturers and associations).
      4. Self-harm-related content defect rate: includes language pertaining to actions intended to hurt, injure, or damage one's body or kill oneself.
      5. Jailbreak defect rate
   2. Generation quality metrics: evaluate the overall quality and coherence of the generated content.
      1. Coherence: measures how well the language model can produce output that flows smoothly, reads naturally, and resembles human-like language
         1. When to use: when assessing the readability and user-friendliness of your model's generated responses in real-world applications.
      2. Fluency: measures the grammatical proficiency of a generative AI's predicted answer.
         1. When to use: when evaluating the linguistic correctness of the AI-generated text, ensuring that it adheres to proper grammatical rules, syntactic structures, and vocabulary usage in the generated responses.
      3. Groundedness: measures how well the model's generated answers align with information from the source data (user-defined context).
         1. When to use: need to verify that AI-generated responses align with and are validated by the provided context.
      4. Relevance: measures the extent to which the model's generated responses are pertinent and directly related to the given questions.
         1. When to use: when evaluating the AI system's performance in understanding the input and generating contextually appropriate responses.
      5. Retrieval score: measures the extent to which the model's retrieved documents are pertinent and directly related to the given questions.
         1. When to use: want to guarantee that the documents retrieved are highly relevant for answering your users' questions. This score helps ensure the quality and appropriateness of the retrieved content.
      6. Similarity: measures the similarity between a source data (ground truth) sentence and the generated response by an AI model.
         1. When to use: want an objective evaluation of an AI model's performance, particularly in text generation tasks where you have access to ground truth responses. GPT-similarity enables you to assess the generated text's semantic alignment with the desired content, helping to gauge the model's quality and accuracy.

**Google Cloud (Vertex AI)**

1. Computation-based metrics: compare whether the LLM-generated results are consistent with a ground-truth dataset of input and output pairs.
   1. Lexicon-based metrics: Use math to calculate the string similarities between LLM-generated results and ground truth, such as Exact Match and ROUGE.
   2. Count-based metrics: Aggregate the number of rows that hit or miss certain ground-truth labels, such as F1-score, Accuracy, and Tool Name Match.
   3. Embedding-based metrics: Calculate the distance between the LLM-generated results and ground truth in the embedding space, reflecting their level of similarity.
2. Model-based metrics:
   1. Summarization:
      1. summarization\_quality: describes the model's ability to summarize text.
      2. summarization\_helpfulness: describes the model's ability to satisfy a user's query by summarizing the relevant details in the original text without significant loss in important information.
      3. summarization\_verbosity: measures if a summary is too long or too short.
   2. Question answering:
      1. question\_answering\_quality: describes the model's ability to answer questions given a body of text to reference.
      2. QuestionAnsweringHelpfulness: describes the model's ability to provide important details when answering a question.
      3. QuestionAnsweringCorrectness: describes the model's ability to correctly answer a question.
      4. QuestionAnsweringRelevance: describes the model's ability to respond with relevant information when asked a question.
   3. Tool use and function calling:
      1. tool\_call\_valid: describes the model's ability to predict a valid tool call. Only the first tool call is inspected.
      2. ToolNameMatch: describes the model's ability to predict a tool call with the correct tool name. Only the first tool call is inspected.
      3. ToolParameterKeyMatch: describes the model's ability to predict a tool call with the correct parameter names.
      4. ToolParameterKVMatch: describes the model's ability to predict a tool call with the correct parameter names and key values.
   4. General text generation:
      1. exact\_match: computes whether a prediction parameter matches a reference parameter exactly.
      2. bleu (BiLingual Evaluation Understudy): holds the result of an algorithm for evaluating the quality of the prediction, which has been translated from one natural language to another natural language. The quality of the prediction is considered to be the correspondence between a prediction parameter and its reference parameter.
      3. rouge: is used to compare the provided prediction parameter against a reference parameter. All rouge metrics return the F1 score. rougeLsum is calculated by default, but you can specify the rouge variant that you want to use.
      4. coherence: describes the model's ability to provide a coherent response.
      5. fluency: describes the model's language mastery.
      6. safety: describes the model's level of safety, that is, whether the response contains any unsafe text.
      7. groundedness: describes the model's ability to provide or reference information included only in the input text.
      8. fulfillment: describes the model's ability to fulfill instructions.

<https://cloud.google.com/vertex-ai/generative-ai/docs/models/determine-eval>

**UXUI Metrics and KPIs**

|  |  |
| --- | --- |
| KPI | Metrics |
| A KPI is a measurement of success for your business. | UX metrics help you determine how well users are interacting with your brand and products. |

KPIs:

1. Net Promoter Score (NPS): gauge how satisfied someone was with their overall experience using one of your products or services
2. Customer satisfaction (CSAT): how well a product or service fulfills customers’ expectations.
3. Conversion Rate: how many people or new visitors complete a task such as signing up, completing an order, making a purchase, etc. (in our case: sign up -> start conversation?)
4. Time-on-task: how long users spend on certain tasks or activities. (in our case: user engagement)

Metrics:

1. Task success rate: whether or not a user is able to complete the task they have been given.
2. Error rate: the number of times a user makes a mistake while trying to complete a task or reach a goal. You can measure it in two ways:
   1. The first option is to measure all of the errors. This is usually just called the total error rate.
   2. The second option is to focus on one error and see how frequently it occurs. This is known as the average error occurrence rate.