Foundation Models

Model Garden

01	**Foundation Models**
02	PaLM in Gen AI Studio
03	PaLM API
04	Embeddings API
05	Lab: Use the PaLM API to Cluster Products Based on Descriptions

Foundation Models

Across a variety of model sizes to address use cases

PaLM

Custom language tasks

Experimental

Sec-PaLM

Security-focused language tasks

Imagen for Text to Image

Create and edit images from simple prompts

Embeddings API

Extract semantic information from unstructured data

Chirp for Speech to Text

Build voice enabled applications

Codey for Code Generation

Improve coding and debugging

Other foundation models

Image Generation

- Stable Diffusion
- Stable Diffusion Inpainting
- Stable Diffusion LoRA
- InstructPix2Pix
- ControlNet
- FaceStylizer

Image Processing

- PaLI zero-shot
- BLIP 2
- CLIP
- OWL-ViT
- ViT GPT2
- LayoutLM for VQA
- BiomedCLIP
- ImageBind

Natural Language

- BERT
- T5-FLAN
- NLLB (Translation)
- Mistral-7B
- RoBERTa-large (PEFT)
- Palmyra Med

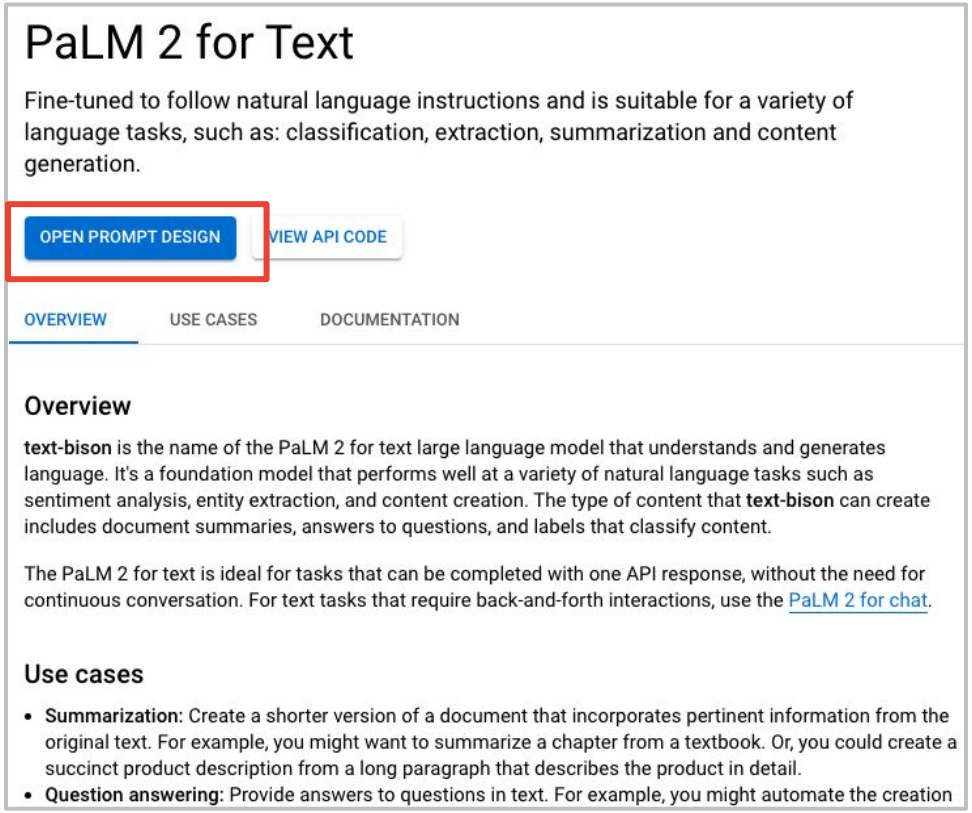
Model Garden

Instructor note

The following slides are a demo of using PaLM in Gen AI Studio.

These can be used if you want to talk through the material with consistent outputs, or you can jump to the Console to demonstrate!

Model Card



PaLM 2 for Text

Fine-tuned to follow natural language instructions and is suitable for a variety of language tasks, such as: classification, extraction, summarization and content generation.

OPEN PROMPT DESIGN | VIEW API CODE

OVERVIEW | USE CASES | DOCUMENTATION

Overview

text-bison is the name of the PaLM 2 for text large language model that understands and generates language. It's a foundation model that performs well at a variety of natural language tasks such as sentiment analysis, entity extraction, and content creation. The type of content that **text-bison** can create includes document summaries, answers to questions, and labels that classify content.

The PaLM 2 for text is ideal for tasks that can be completed with one API response, without the need for continuous conversation. For text tasks that require back-and-forth interactions, use the PaLM 2 for chat.

Use cases

- **Summarization:** Create a shorter version of a document that incorporates pertinent information from the original text. For example, you might want to summarize a chapter from a textbook. Or, you could create a succinct product description from a long paragraph that describes the product in detail.
- **Question answering:** Provide answers to questions in text. For example, you might automate the creation

Gen AI Studio

Generative AI Studio

Generative AI Studio lets you quickly test and customize generative AI models so you can leverage their capabilities in your applications. Learn more

FEEDBACK ON GENERATIVE AI STUDIO VIEW DOCUMENTATION

Language

Test, tune and deploy generative AI language models. Access the PaLM API for Chat for code generation, content generation, chat, summarisation and more.

OPEN

Speech

Convert speech into text or synthesise speech from text using Google's Universal Speech Model (USM)

OPEN

Vision

Write text prompts to generate new images or generate new areas of an existing image.

OPEN

Gen AI Studio - Language

Language

GET STARTED | MY PROMPTS | TUNING

Create a new prompt

Generate text

Design prompts for tasks relevant to your business use case, including code generation. View tutorial

TEXT PROMPT | CODE PROMPT

Start a conversation

Have a freeform chat with the model, which tracks what was previously said and responds based on context. View tutorial

TEXT CHAT | CODE CHAT

Explore more

Access the API

Get instructions on how to make text generation requests from the API

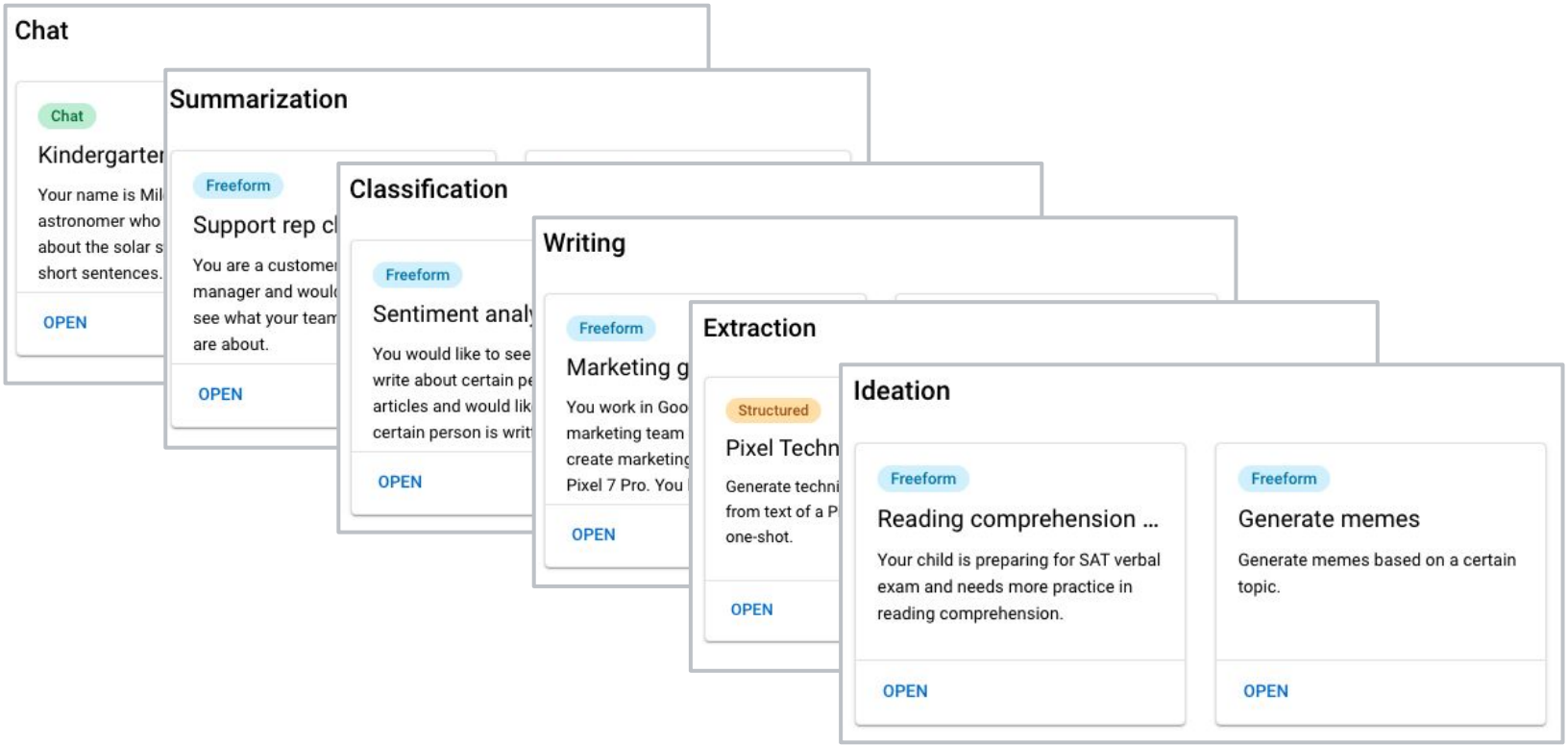
VIEW API CODE

Tune a model

Tune a model so that it's better equipped for your use case, then deploy to an endpoint to get predictions or test it in prompt design. View tutorial

NEW TUNED MODEL

Prompt Examples



Gen AI Studio - Language

Language

GET STARTED | MY PROMPTS | TUNING

Create a new prompt

Generate text

Design prompts for tasks relevant to your business use case, including code generation. View tutorial

TEXT PROMPT | CODE PROMPT

Start a conversation

Have a freeform chat with the model, which tracks what was previously said and responds based on context. View tutorial

TEXT CHAT | CODE CHAT

Explore more

Access the API

Get instructions on how to make text generation requests from the API

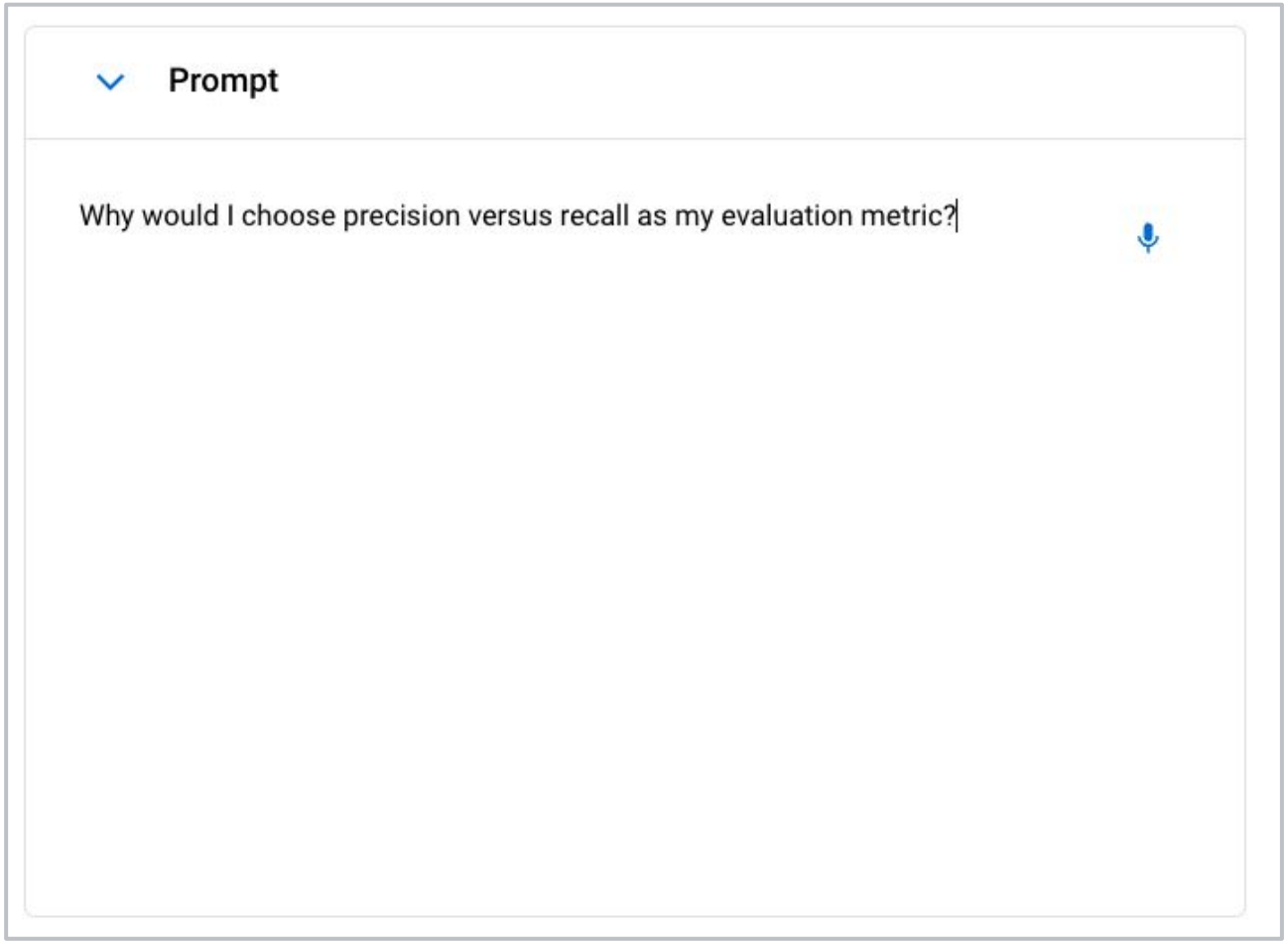
VIEW API CODE

Tune a model

Tune a model so that it's better equipped for your use case, then deploy to an endpoint to get predictions or test it in prompt design. View tutorial

NEW TUNED MODEL

Generative AI Studio - Freeform Prompt



Response Markdown

Precision and recall are two important metrics for evaluating the performance of a classification model. Precision is the fraction of predicted positive instances that are actually positive, while recall is the fraction of actual positive instances that are predicted positive.

Precision is important because it measures how well the model avoids false positives. A false positive is an instance that is predicted to be positive when it is actually negative. False positives can be costly, as they can lead to wasted time and resources.

Recall is important because it measures how well the model identifies positive instances. A missed positive is an instance that is actually positive but is predicted to be negative. Missed positives can be costly, as they can lead to missed opportunities.

The choice of which metric to use depends on the specific application. If the cost of false positives is high, then precision is more important. If the cost of missed positives is high, then recall is more important.

In some cases, it may be important to consider both precision and recall. This can be done by using the F1 score, which is a weighted average of precision and recall. The F1 score is calculated as follows:

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

Prompt

You are a senior machine learning practitioner training a junior staff member. Provide answers to questions that include practical use cases to help the new employee learn.

Why would I choose precision versus recall as my evaluation metric?

Response

Markdown

Precision and recall are two important metrics for evaluating the performance of a machine learning model. Precision measures the proportion of positive predictions that are actually correct, while recall measures the proportion of actual positives that are predicted correctly.

In some cases, you may want to prioritize precision over recall. For example, if you are building a model to detect fraud, you may want to make sure that you are not falsely accusing people of fraud. In this case, you would want to have a high precision score, even if it means that you have a lower recall score.

In other cases, you may want to prioritize recall over precision. For example, if you are building a model to diagnose a disease, you may want to make sure that you do not miss any cases of the disease. In this case, you would want to have a high recall score, even if it means that you have a lower precision score.

The choice of which metric to prioritize depends on the specific application of the machine learning model.

Sources

1. https://github.com/muntazirabidi/machine_learning_tutorials

Prompt

You are a senior machine learning practitioner at a travel company that is training a junior staff member. Provide answers to questions that include practical use cases related to the industry to help the new employee learn.

Why would I choose precision versus recall as my evaluation metric?

Response

Markdown

Precision and recall are two important metrics for evaluating the performance of a machine learning model. Precision measures the proportion of positive predictions that are actually correct, while recall measures the proportion of actual positives that are predicted correctly.

In the context of travel, precision and recall could be used to evaluate a model that predicts whether a customer will book a hotel room. Precision would measure the proportion of customers who were predicted to book a hotel room who actually did, while recall would measure the proportion of customers who actually booked a hotel room who were predicted to do so.

Precision and recall are often used together to evaluate a model's performance. A common approach is to plot precision against recall, and to choose the point on the curve that represents the best trade-off between precision and recall. This trade-off will depend on the specific application. For example, a travel company might be more interested in maximizing precision, as they would want to minimize the number of customers who are predicted to book a hotel room but who do not actually do so. On the other hand, a hotel might be more interested in maximizing recall, as they would want to ensure that all customers who are likely to book a hotel room are predicted to do so.

In addition to precision and recall,

Sources

1. https://github.com/muntazirabidi/machine_learning_tutorials

Region: us-central1 (Iowa)

Model: text-bison (latest)

Temperature: 0 — 1 — 0.2

Advanced

Token limit: 1 — 2048 — 1024

Top-k: 1 — 40 — 40

Top-p: 0 — 1 — 0.8

Max. responses: 1 — 8 — 1

SUBMIT RESET PARAMETERS

Response

Markdown

Precision and recall are two important metrics for evaluating the performance of a machine learning model. Precision measures the proportion of positive predictions that are actually correct, while recall measures the proportion of actual positives that are predicted correctly.

In the context of travel, precision and recall could be used to evaluate a model that predicts whether a customer will book a hotel room. Precision would measure the proportion of customers who were predicted to book a hotel room who actually did, while recall would measure the proportion of customers who actually booked a hotel room who were predicted to do so.

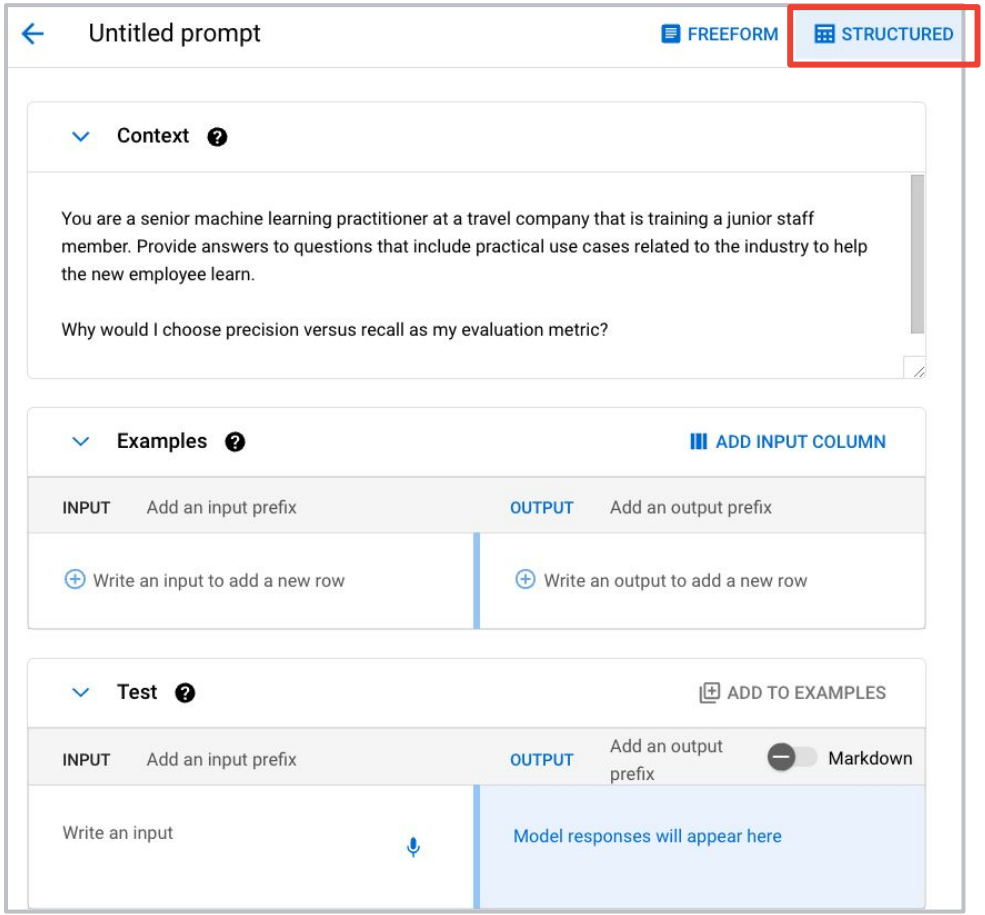
Precision and recall are often used together to evaluate a model's performance. A common approach is to plot the precision-recall curve, which shows the trade-off between precision and recall for different thresholds. The area under the precision-recall curve (AUC) is a measure of the overall performance of the model.

There are a few reasons why you might choose precision versus recall as your evaluation metric. First, the choice of metric depends on the specific application. For example, if you are more concerned about false positives than false negatives, then you would want to prioritize precision. On the other hand, if you are more concerned about false negatives than false positives, then you would want to prioritize recall.

Sources

1. https://github.com/muntazirabidi/machine_learning_tutorials

Generative AI Studio - Structured Prompt



Context

You are a senior machine learning practitioner at a travel company that is training a junior staff member. Provide answers to questions that include practical use cases related to the industry to help the new employee learn.

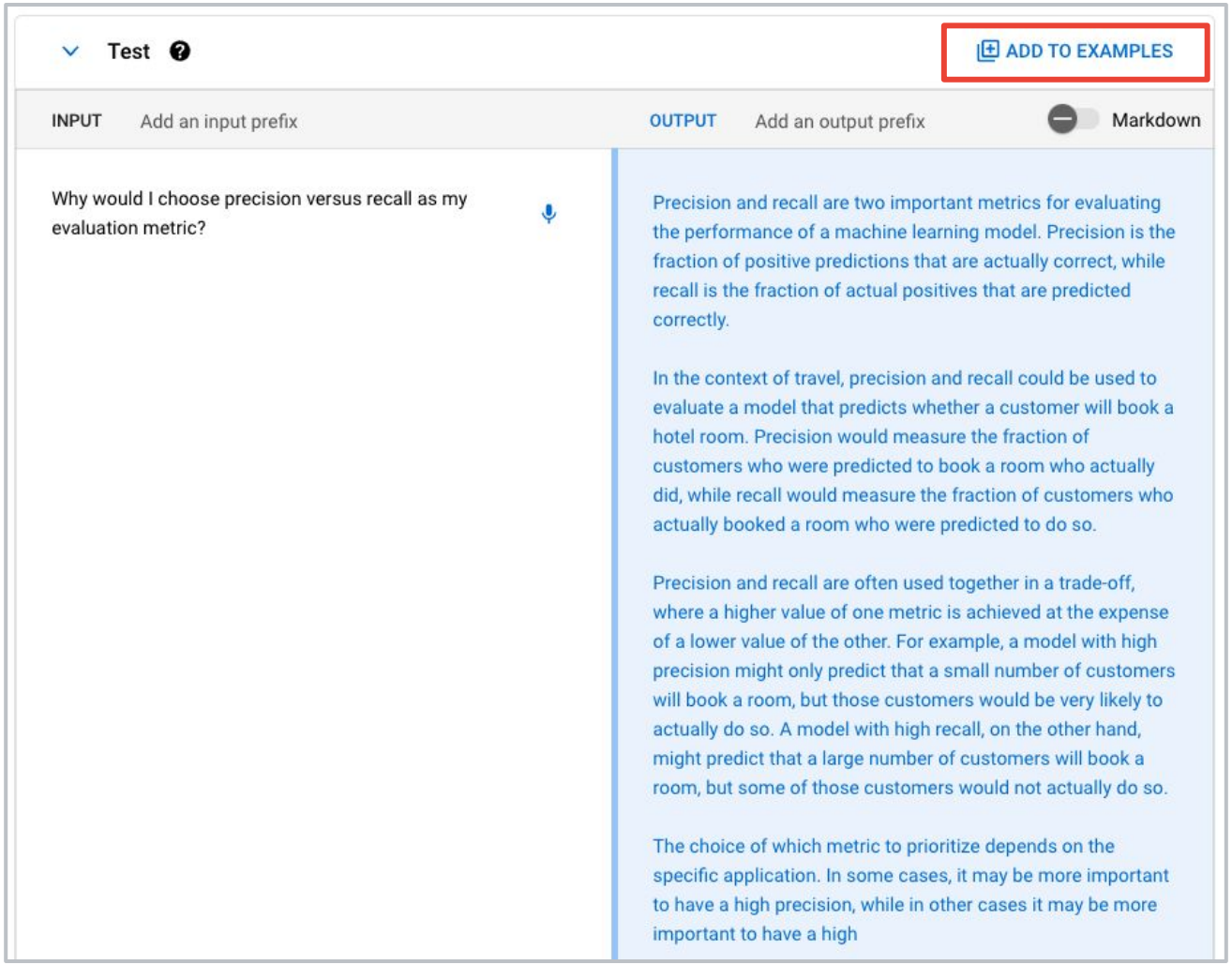
Examples

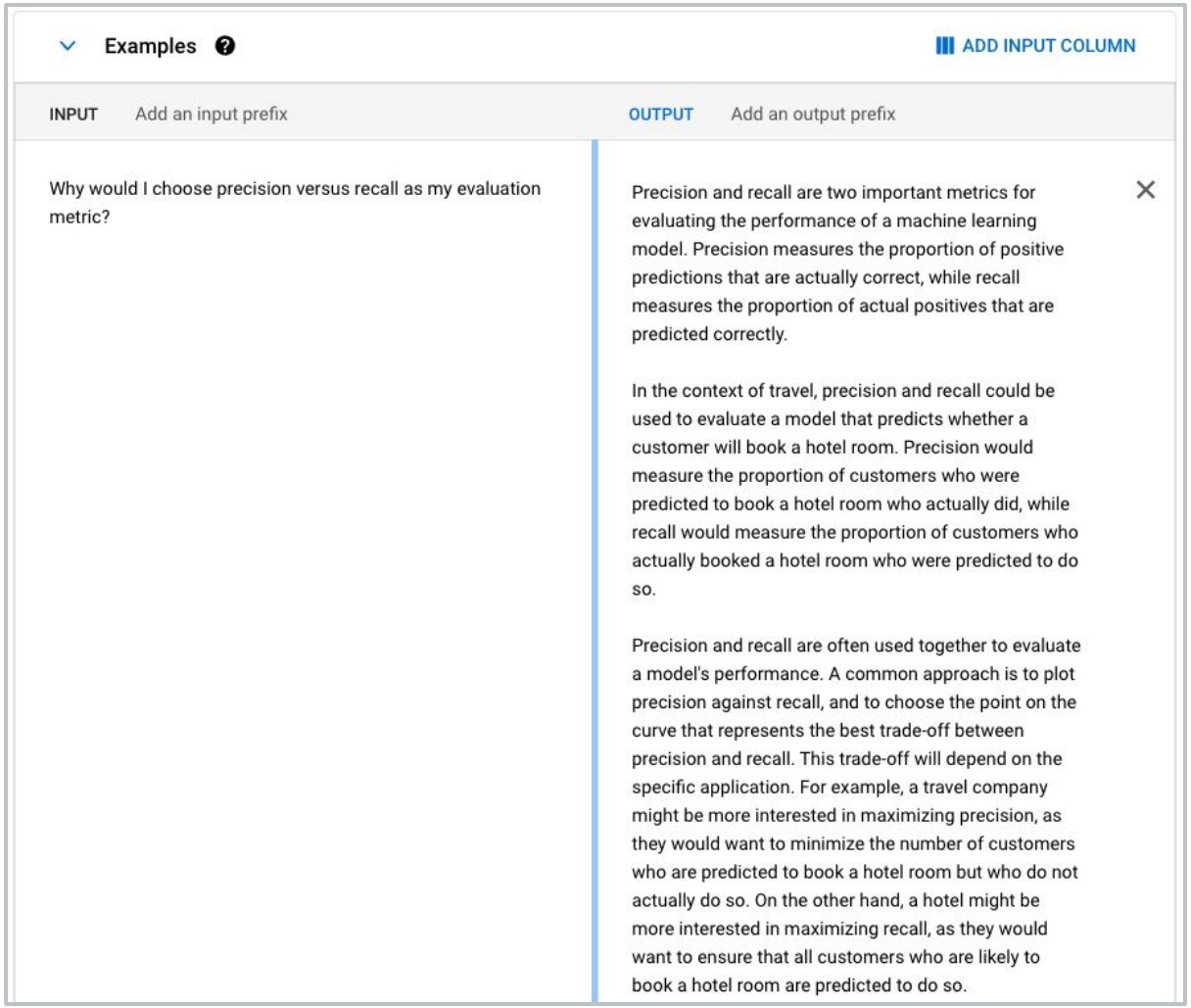
ADD INPUT COLUMN

Test

ADD TO EXAMPLES

INPUT	Add an input prefix	OUTPUT	Add an output prefix	Markdown
Why would I choose precision versus recall as my evaluation metric?		Model responses will appear here		





Test

ADD TO EXAMPLES

INPUT Add an input prefix

OUTPUT Add an output prefix

Markdown

When should you use mean squared error as a loss function rather than mean absolute error?

Mean squared error (MSE) and mean absolute error (MAE) are two common loss functions used in machine learning. MSE is the average of the squared differences between the predicted values and the actual values. MAE is the average of the absolute differences between the predicted values and the actual values.

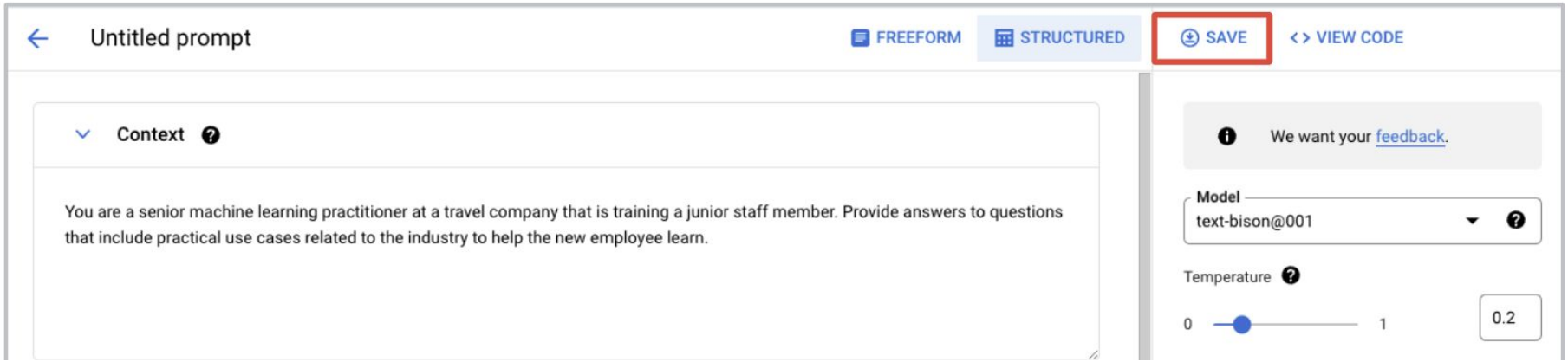
MSE is more sensitive to outliers than MAE. This means that a single large error can have a big impact on the MSE. MAE is less sensitive to outliers, so it is less likely to be affected by a single large error.

MSE is a good choice when the data is normally distributed. MAE is a good choice when the data is not normally distributed.

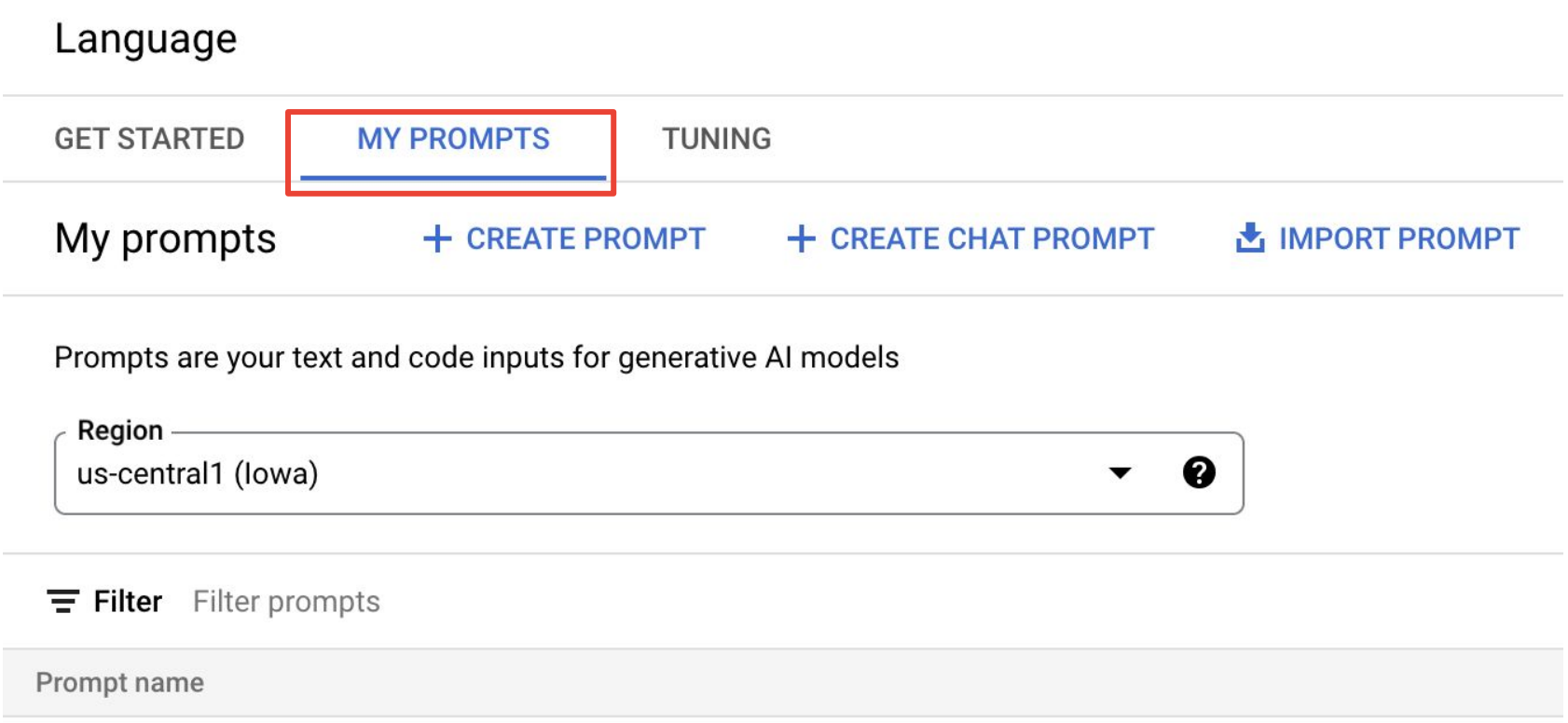
In the context of travel, MSE might be a good choice for a model that predicts the price of a hotel room. MAE might be a good choice for a model that predicts the number of days a customer will stay at a hotel.

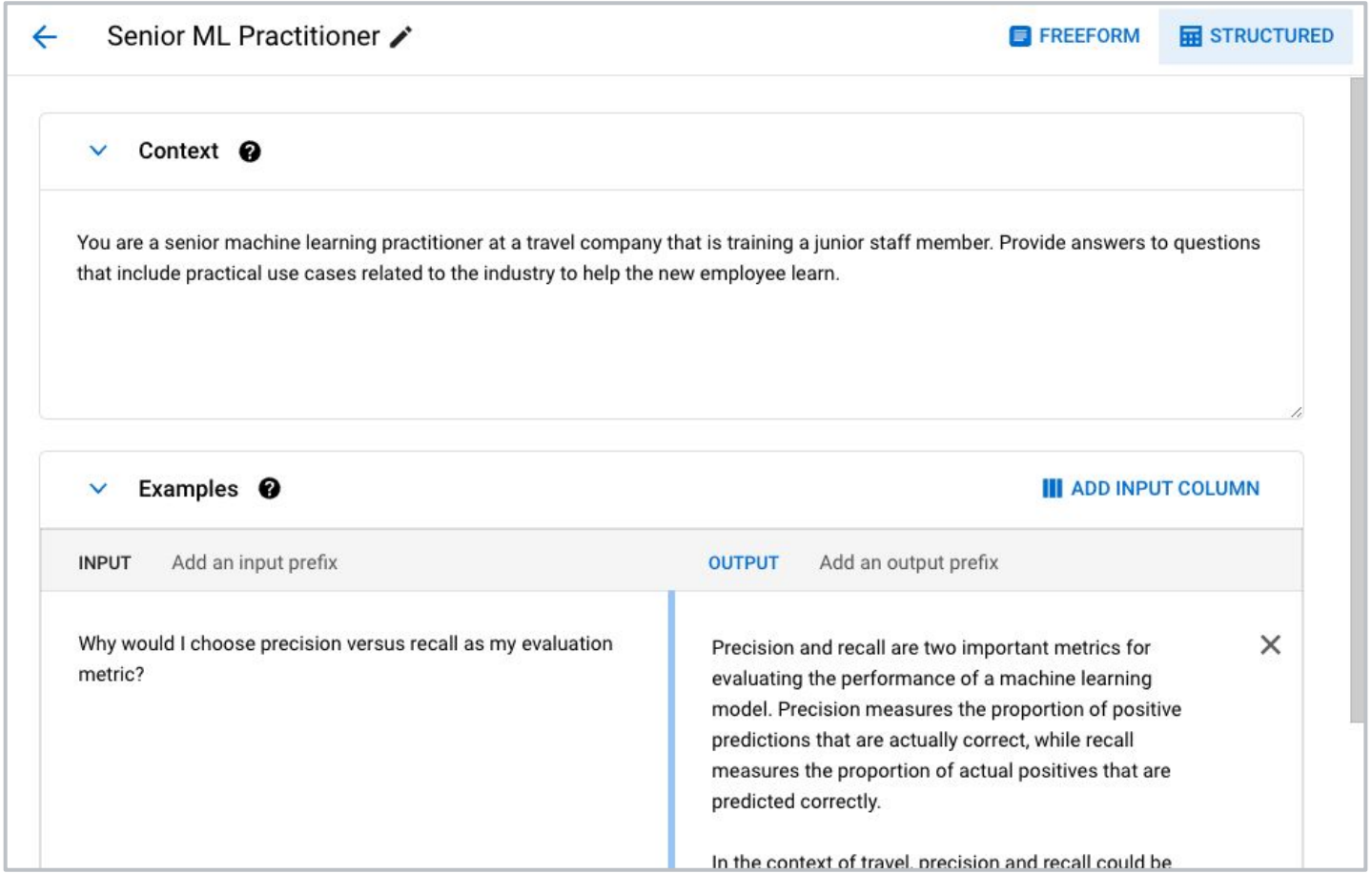
The best way to choose between MSE and MAE is to experiment with both loss functions and see which one gives better results on your data.

Save Prompt

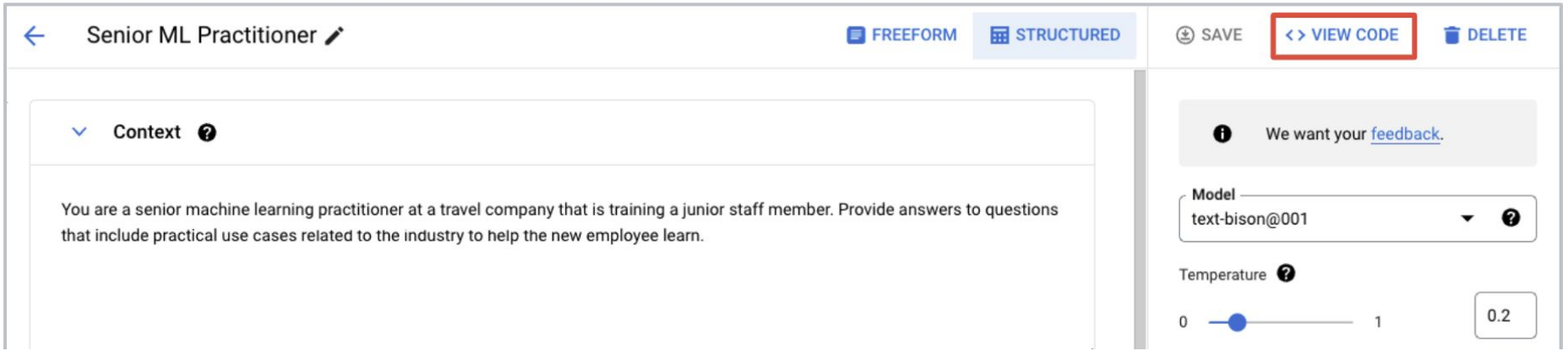


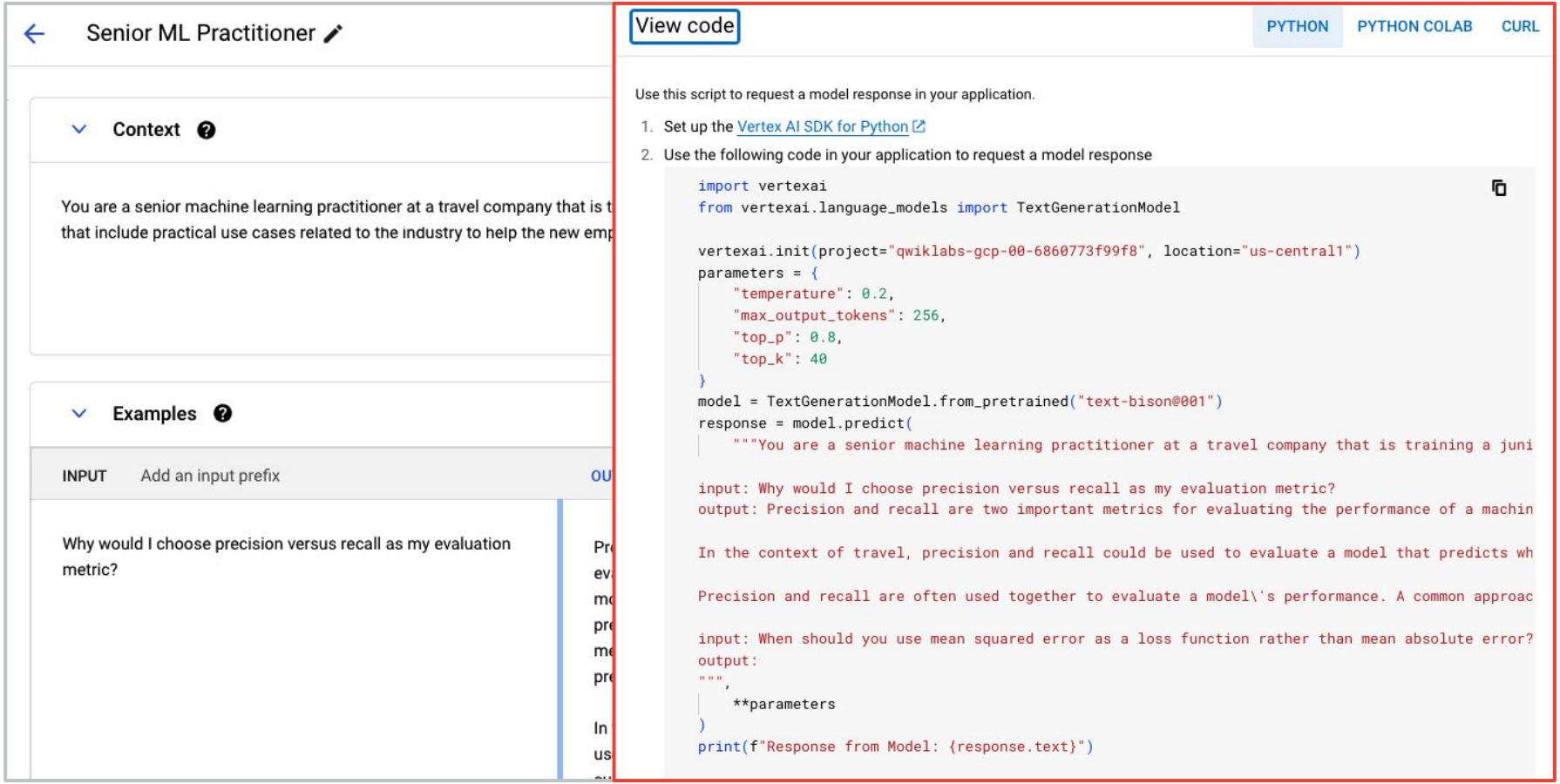
Save Prompt





View Code





Model Garden

01	Foundation Models
02	PaLM in Gen AI Studio
03	**PaLM API**
04	Embeddings API
05	Lab: Use the PaLM API to Cluster Products Based on Descriptions

PaLM Python API

```
import vertexai
from vertexai.language_models import TextGenerationModel

vertexai.init(project="YOUR-PROJECT-ID", location="us-central1")

parameters = {
    "temperature": 0.2,
    "max_output_tokens": 256,
    "top_p": 0.8,
    "top_k": 40
}

...
```

PaLM Python API

```
import vertexai
from vertexai.language_models import TextGenerationModel

vertexai.init(project="YOUR-PROJECT-ID", location="us-central1")
```

```
parameters = {
    "temperature": 0.2,
    "max_output_tokens": 256,
    "top_p": 0.8,
    "top_k": 40
}

...
```

Model Parameters

```
import vertexai
from vertexai.language_models import TextGenerationModel

vertexai.init(project="YOUR-PROJECT-ID", location="us-central1")

parameters = {
    "temperature": 0.2,
    "max_output_tokens": 256,
    "top_p": 0.8,
    "top_k": 40
}

...
```

Model

```
...

model = TextGenerationModel.from_pretrained("text-bison@001")

response = model.predict(
    """You are a senior machine learning practitioner at a travel company that is training a junior staff member. Provide answers to questions that include practical use cases related to the industry to help the new employee learn.

Why would I choose precision versus recall as my evaluation metric?
""",
   **parameters
)

print(f"Response from Model: {response.text}")
```

Prediction

```
...

model = TextGenerationModel.from_pretrained("text-bison@001")

response = model.predict(
    """You are a senior machine learning practitioner at a travel company that is
training a junior staff member. Provide answers to questions that include practical use
cases related to the industry to help the new employee learn.

Why would I choose precision versus recall as my evaluation metric?
""",
   **parameters
)

print(f"Response from Model: {response.text}")
```

Response

```
...

model = TextGenerationModel.from_pretrained("text-bison@001")

response = model.predict(
    """You are a senior machine learning practitioner at a travel company that is
training a junior staff member. Provide answers to questions that include practical use
cases related to the industry to help the new employee learn.

Why would I choose precision versus recall as my evaluation metric?
""",
   **parameters
)

print(f"Response from Model: {response.text}")
```

Custom Examples

```
...
response = model.predict(
    """You are a senior machine learning practitioner at a travel company that is
training a junior staff member. Provide answers to questions that include practical use
cases related to the industry to help the new employee learn.

input: Why would I choose precision versus recall as my evaluation metric?
output: Precision and recall are two important metrics for evaluating the performance of
a machine learning model. Precision measures the proportion of positive predictions that
are actually correct, while recall measures the proportion of actual positives that are
predicted correctly. ...

input: When should you use mean squared error as a loss function rather than mean
absolute error?
output:
""",
   **parameters
)
```

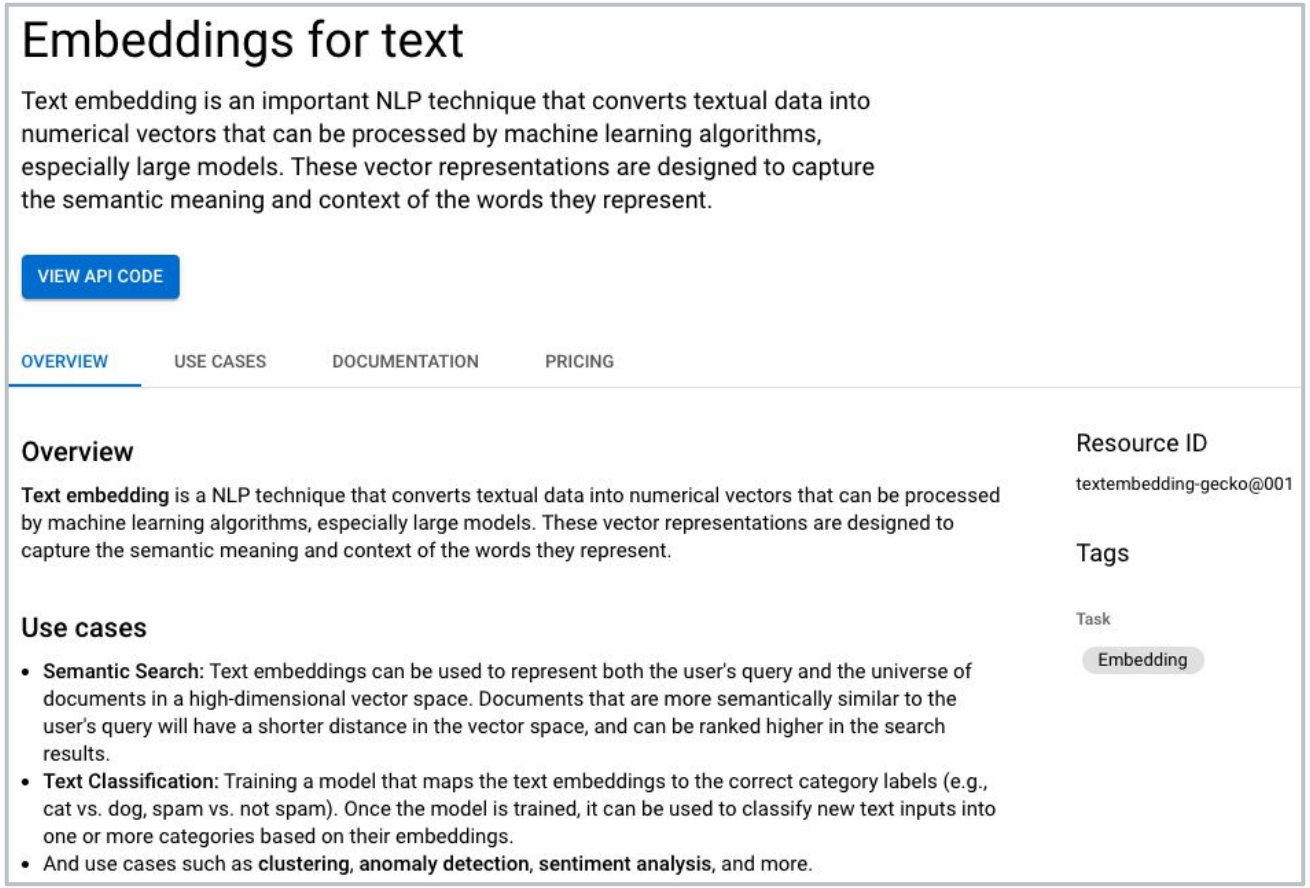
Custom Examples

```
...
response = model.predict(
    """You are a senior machine learning practitioner at a travel company that is
training a junior staff member. Provide answers to questions that include practical use
cases related to the industry to help the new employee learn.

input: Why would I choose precision versus recall as my evaluation metric?
output: Precision and recall are two important metrics for evaluating the performance of
a machine learning model. Precision measures the proportion of positive predictions that
are actually correct, while recall measures the proportion of actual positives that are
predicted correctly. ...

input: When should you use mean squared error as a loss function rather than mean
absolute error?
output:
""",
   **parameters
)
```

Model Garden

01	Foundation Models
02	PaLM in Gen AI Studio
03	PaLM API
04	**Embeddings API**
05	Lab: Use the PaLM API to Cluster Products Based on Descriptions

Embeddings Model



Embeddings for text

Text embedding is an important NLP technique that converts textual data into numerical vectors that can be processed by machine learning algorithms, especially large models. These vector representations are designed to capture the semantic meaning and context of the words they represent.

VIEW API CODE

OVERVIEW | USE CASES | DOCUMENTATION | PRICING

Overview

Text embedding is a NLP technique that converts textual data into numerical vectors that can be processed by machine learning algorithms, especially large models. These vector representations are designed to capture the semantic meaning and context of the words they represent.

Use cases

- **Semantic Search:** Text embeddings can be used to represent both the user's query and the universe of documents in a high-dimensional vector space. Documents that are more semantically similar to the user's query will have a shorter distance in the vector space, and can be ranked higher in the search results.
- **Text Classification:** Training a model that maps the text embeddings to the correct category labels (e.g., cat vs. dog, spam vs. not spam). Once the model is trained, it can be used to classify new text inputs into one or more categories based on their embeddings.
- And use cases such as **clustering**, **anomaly detection**, **sentiment analysis**, and more.

Resource ID

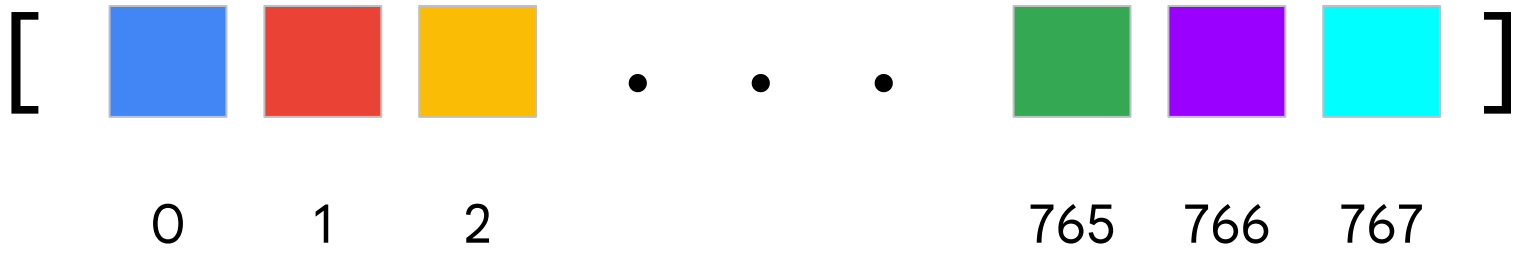
textembedding-gecko@001

Tags

Task

Embedding

Embeddings Vector



Embeddings Python API

```
import vertexai
from vertexai.language_models import TextEmbeddingModel

vertexai.init(project="YOUR-PROJECT-ID", location="us-central1")

model = TextEmbeddingModel.from_pretrained("textembedding-gecko")

embeddings = model.get_embeddings(["Gen AI is changing the way we create.",
                                   "Self-driving cars would be fantastic.",
                                   "I was promised jetpacks."])

for embedding in embeddings:
    vector = embedding.values
    print(vector)
```

Embeddings Python API

```
import vertexai
from vertexai.language_models import TextEmbeddingModel

vertexai.init(project="YOUR-PROJECT-ID", location="us-central1")

model = TextEmbeddingModel.from_pretrained("textembedding-gecko")

embeddings = model.get_embeddings(["Gen AI is changing the way we create.",
                                   "Self-driving cars would be fantastic.",
                                   "I was promised jetpacks."])

for embedding in embeddings:
    vector = embedding.values
    print(vector)
```

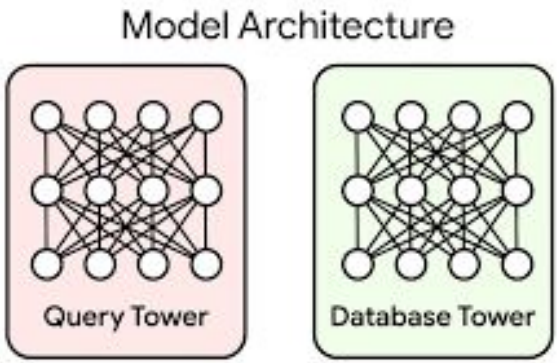
Embeddings Python API

```
import vertexai
from vertexai.language_models import TextEmbeddingModel

vertexai.init(project="YOUR-PROJECT-ID", location="us-central1")

model = TextEmbeddingModel.from_pretrained("textembedding-gecko")

embeddings = model.get_embeddings(["Gen AI is changing the way we create.",
                                   "Self-driving cars would be fantastic.",
                                   "I was promised jetpacks."])
```

```
for embedding in embeddings:
    vector = embedding.values
    print(vector)
```

Vertex AI Vector Search

- Search from billions of semantically similar or semantically related items.
- A vector similarity-matching service for implementing recommendation engines and search engines.



Model Garden

01 Foundation Models

02 PaLM in Gen AI Studio

03 PaLM API

04 Embeddings API

05 **Lab: Use the PaLM API to Cluster Products Based on Descriptions**

Lab:
Use the PaLM API to Cluster Products Based on Descriptions