



Fine-tuning Models

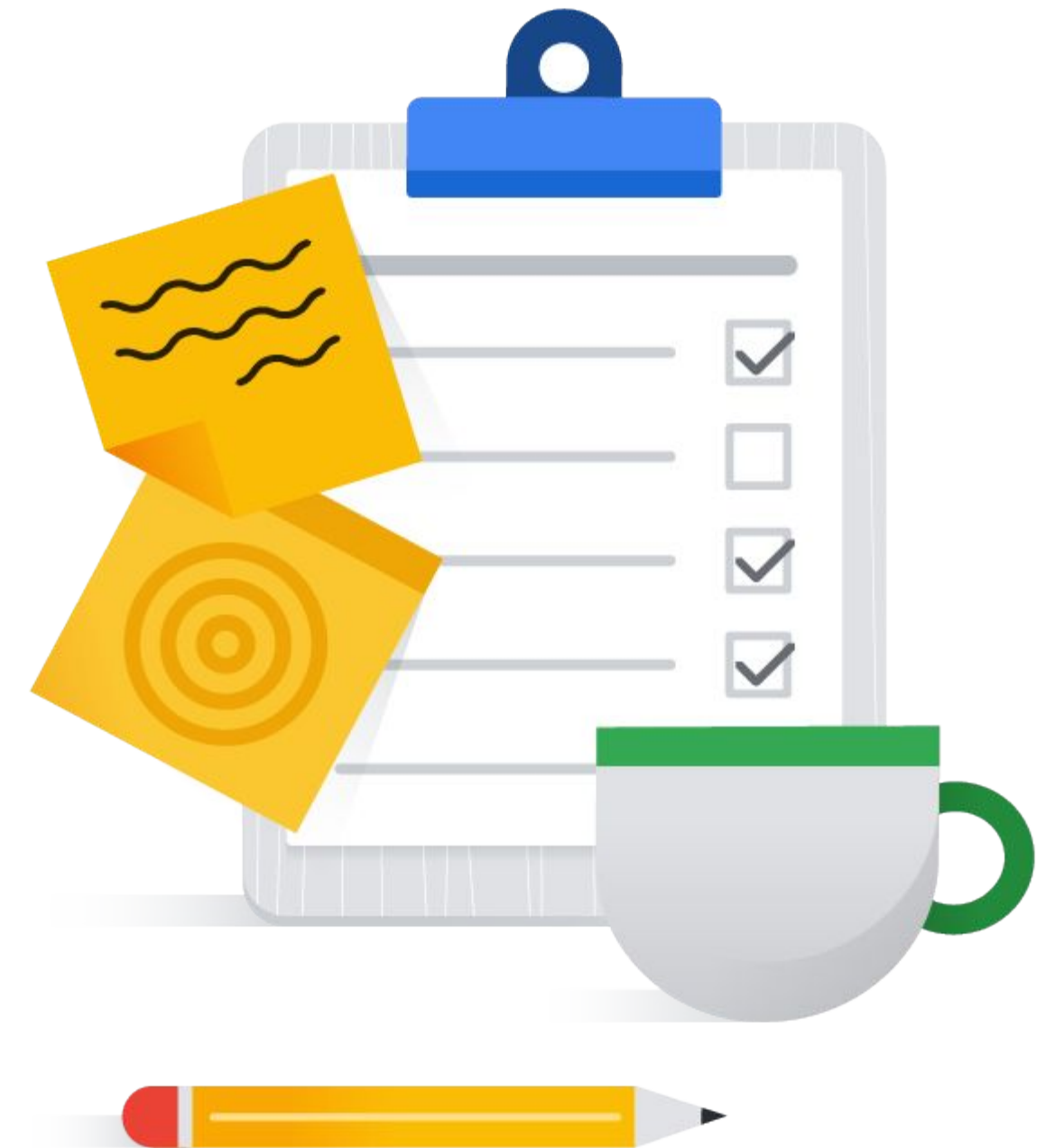
In this module, you learn to ...

- 01 Evaluate scenarios for creating tuned models
- 02 Build workflows for tuning and deploying models
- 03 Use tuned models in your applications



Topics

01	Tuned Models
02	Preparing a Model Tuning Dataset
03	Creating a Tuning Job



Use model tuning to improve model performance on specific tasks

- When few-shot prompting and adding context are not adequate to your use case
- Allows you to teach the model more about what your expected output should be
- You specify a custom dataset which includes prompts along with the expected output
 - Like adding examples, but more of them with custom training
- The custom training jobs learn the outputs (called weights)

Tuning is required when you want output that deviates from general language patterns

- Specific structures or formats for generating output
- Specific behaviors such as when to provide a terse or verbose output
- Specific customized outputs for specific types of inputs

When custom training may be required:

Classification

- Classification with custom classes (groups)
 - Give the model examples, with the correct answers

input_text:

Classify the following text into one of the following classes:

[HR, Sales, Marketing, Customer Service].

Text: Are you currently hiring?

output_text:

HR

When custom training may be required:

Summarization

- Summaries that require specific output
- In the example below, you want to remove personally identifiable information (PII) in a chat summary

input_text:

Summarize:

Jessica: That sounds great! See you in Times Square!

Alexander: See you at 10!

output_text:

#Person1 and #Person2 agree to meet at Times Square at 10:00 AM

When custom training may be required:

Extractive question answering

- The question is about a context and the answer is a substring of the context

input_text:

context: There is evidence that there have been significant changes in Amazon rainforest vegetation over the last 21,000 years through the Last Glacial Maximum (LGM) and subsequent deglaciation.

question: What does LGM stand for?

output_text:

Last Glacial Maximum

Including context in your training data

- In the example below, the **input_text** consists of both a **context** section and a **question** section

input_text:

context: There is evidence that there have been significant changes in Amazon rainforest vegetation over the last 21,000 years through the Last Glacial Maximum (LGM) and subsequent deglaciation.

question: What does LGM stand for?

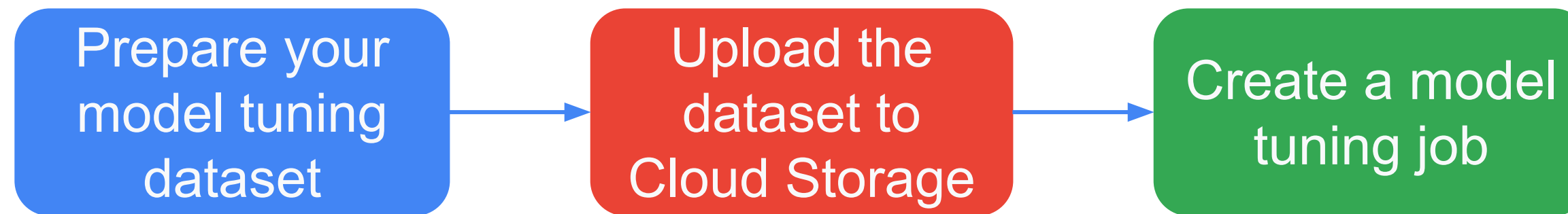
- The context provides additional information for answering the question
- If your training data is formatted in this way, you must format prompts in the same way when using your model for inference
 - I.e. Whatever sections your training data has as input, must be included in prompts in the same order when using the model

Topics

01	Tuned Models
02	Preparing a Model Tuning Dataset
03	Creating a Tuning Job



Model tuning workflow on Vertex AI



- After tuning, the model is automatically deployed to a Vertex AI endpoint using the name you provide in the tuning job
- The model is also available in Vertex AI Studio when creating prompts

Prepare your model tuning dataset

- The training data must be in JSONL format
 - The “L” is for “Line”
 - Each line in the JSONL file is one example
 - It is not an array of objects, it is one object per line
- Each object must have the properties `input_text` and `output_text`

`{}` custom-training-data.jsonl ×

Users > doug > `{}` custom-training-data.jsonl

```
1 {"input_text": "question: How many copies of Gears of War 3 were sold ? context: Like  
2 {"input_text": "question: How many parishes are there in Louisiana ? context: The U .
```

It is important to include instructions into the training data

- The following has no instructions, so it is not a good example

```
{"input_text": "5 stocks to buy now","output_text": "business"}
```

- The following has instructions, so it is a better example

```
{"input_text": "Classify the following text into one of the following classes:  
[business, entertainment] Text: 5 stocks to buy now","output_text": "business"}
```

Including context within the input text

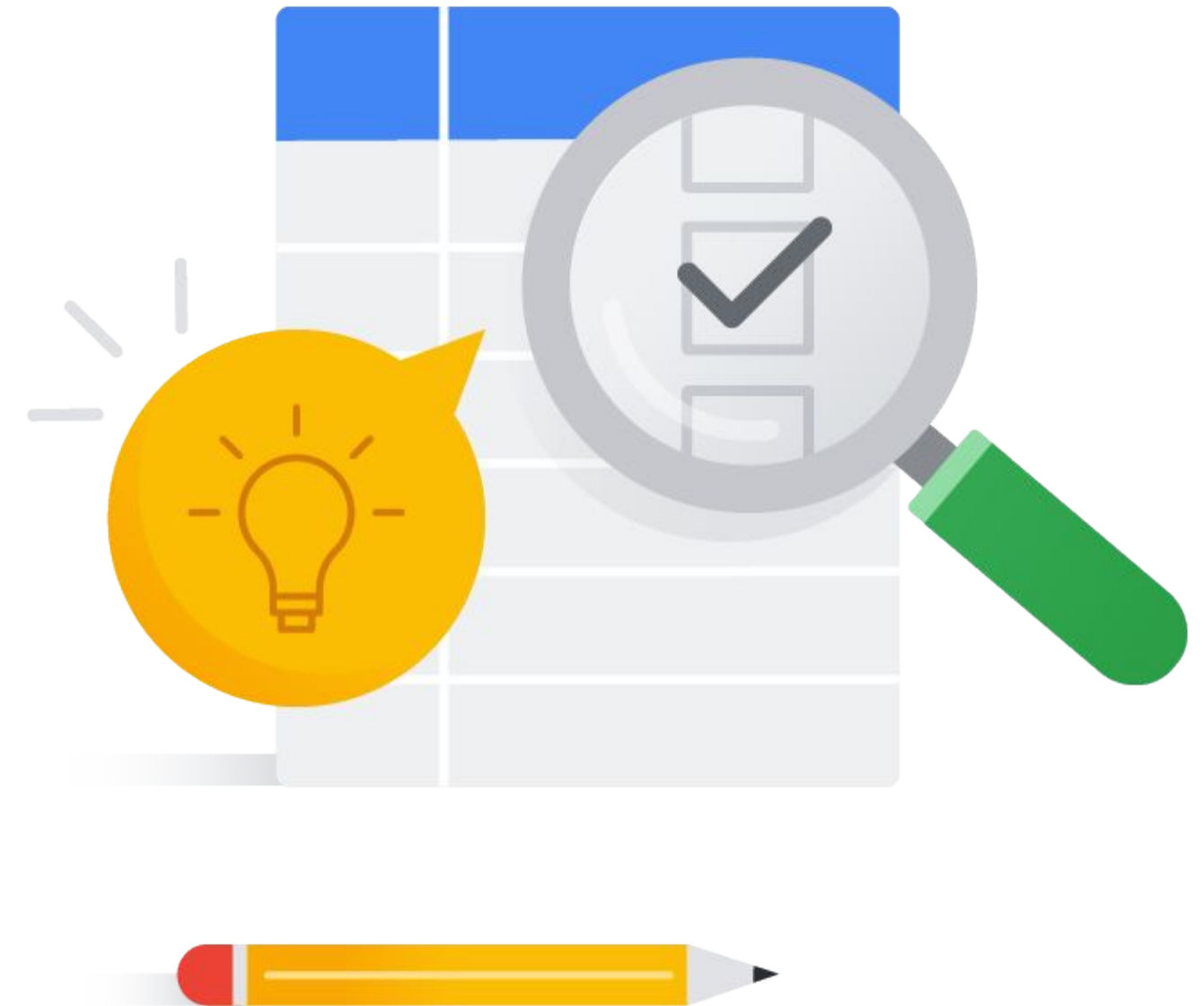
- Notice that the following `input_text` has question and context sections
 - When using the model, remember that prompts need to be formatted the same way
 - Be consistent

```
{"input_text": "question: How many parishes are there in Louisiana? context: The U.S. state of Louisiana is divided into 64 parishes (French: paroisses) in the same manner that 48 other states of the United States are divided into counties, and Alaska is divided into boroughs.", "output_text": "64"}
```

Do Now: Exploring Sample Training Data

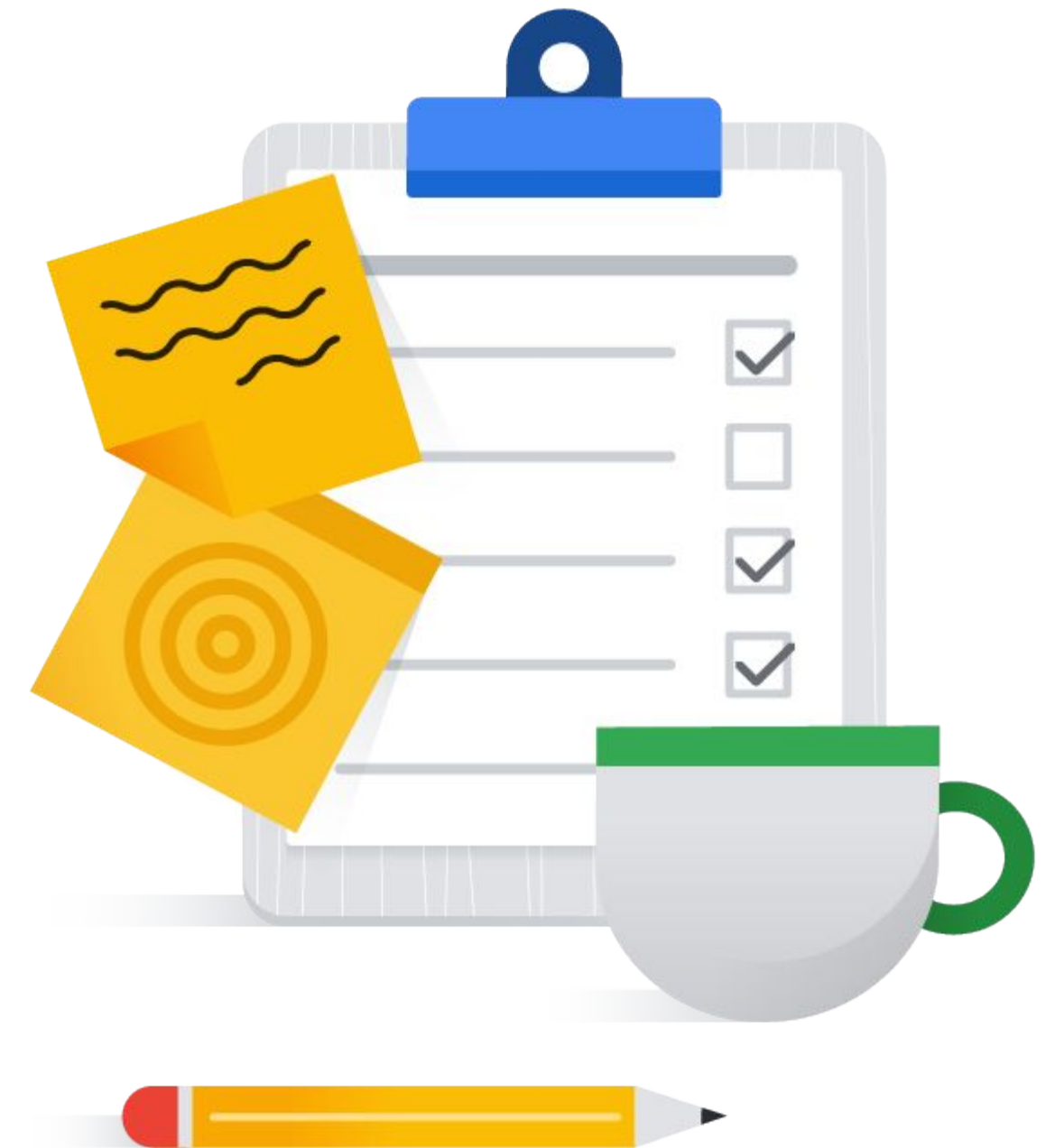
🕒 5 min 🧑🏫

1. Go to:
<https://github.com/roitraining/genai-model-tuning-examples>
2. You will find some example fine-tuning datasets
3. Click on a couple of them and explore the examples
 - a. Each file has 1 example per line
 - b. Each example has input_text and output text attributes



Topics

01	Tuned Models
02	Preparing a Model Tuning Dataset
03	Creating a Tuning Job



Select the Tune a model task in Vertex AI Studio



Tune a model

Tune a model so 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

Specify the location of the data and the job parameters

← Create a tuned model

1 Tuning type

2 Model details

3 Tuning dataset

4 Evaluation (optional)

START TUNING

Choose a tuning method

Tuning improves model quality for a specific domain or dataset. The recommended tuning method depends on the data you have available, your goals and use case. [Learn more about tuning](#)

☒ Supervised tuning

Uses example prompt and model responses to tune the model

☐ Reinforcement learning from human feedback (RLHF)

Uses human preference data to create a separate reward model, which then tunes the foundation model using reinforcement learning

CONTINUE

← Create a tuned model

✓ Tuning type

2 Model details

3 Tuning dataset

4 Evaluation (optional)

START TUNING

Model name *

The name of the new model. Up to 128 characters.

Tuning settings

Base model

text-bison@001

The base model that will be used to create a new tuned model.

Train steps *

300

Learning rate multiplier *

1

The real learning rate is defined as the product of the learning rate multiplier and the underlying recommended learning rate.

gs:// Working directory *

BROWSE

The Cloud Storage location where the artifacts are stored during the pipeline tuning run.

Accelerator type

GPU (us-central1)

The accelerator used for model tuning. The accelerator type you choose determines the region where the model tuning computation occurs.

Region

us-central1 (Iowa)

Where your pipeline tuning job runs and the tuned model is deployed. Your tuning data always remains in this region

Add a TensorBoard instance

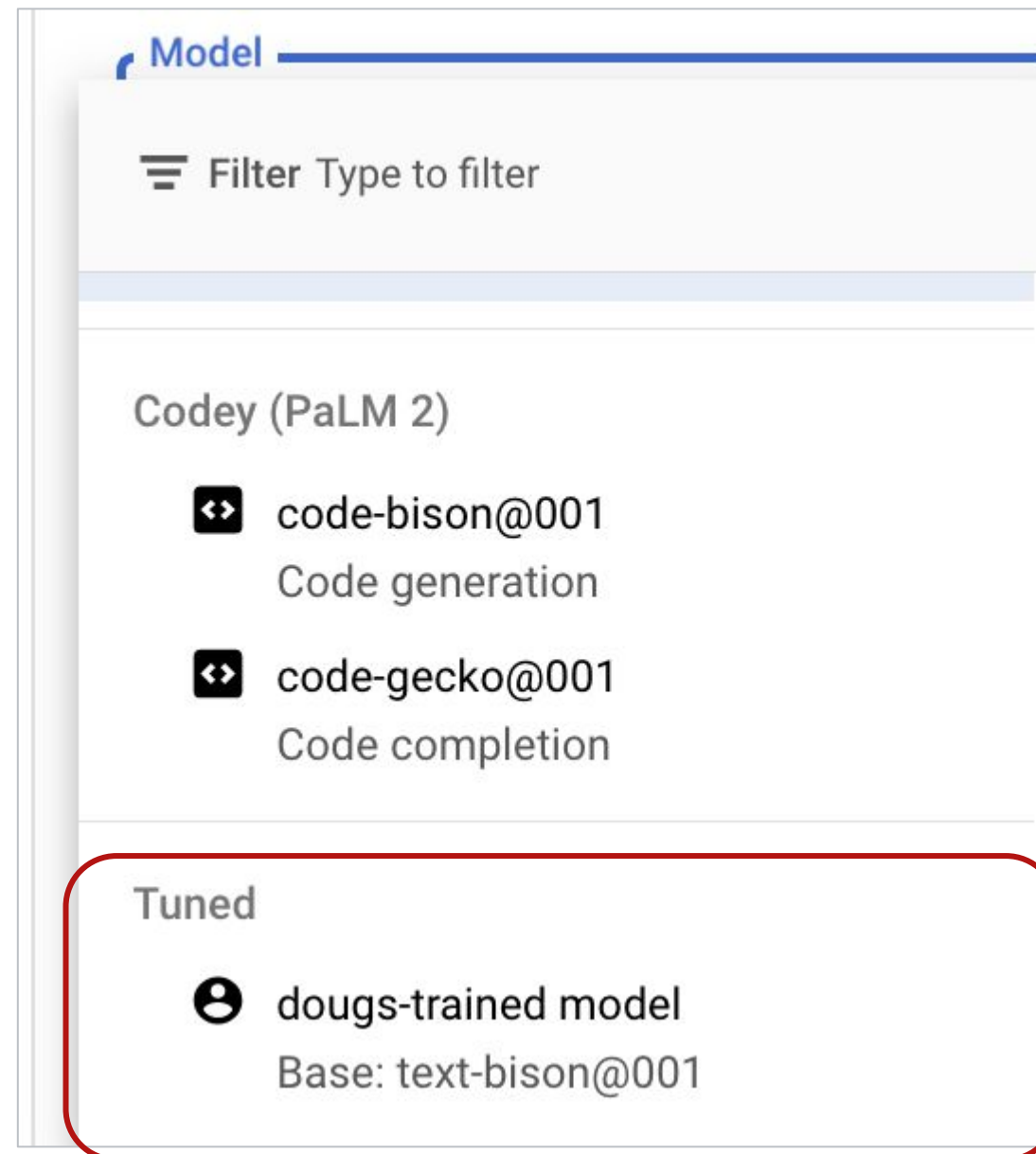
TensorBoard instance

ADVANCED OPTIONS

The number of training examples and training steps needed depends on the task

Task	Suggested # of examples	Training steps
Classification	100+	100-500
Summarization	100-500+	200-1000
Extractive QA	100+	100-500

Tuned models are available from Vertex AI Studio



Vertex AI Studio will generate the code for using tuned models

View code PYTHON PYTHON COLAB CURL

Use this script to request a model response in your application.

1. Set up the [Vertex AI SDK for Python](#)
2. Use the following code in your application to request a model response

```
import vertexai
from vertexai.preview.language_models import TextGenerationModel

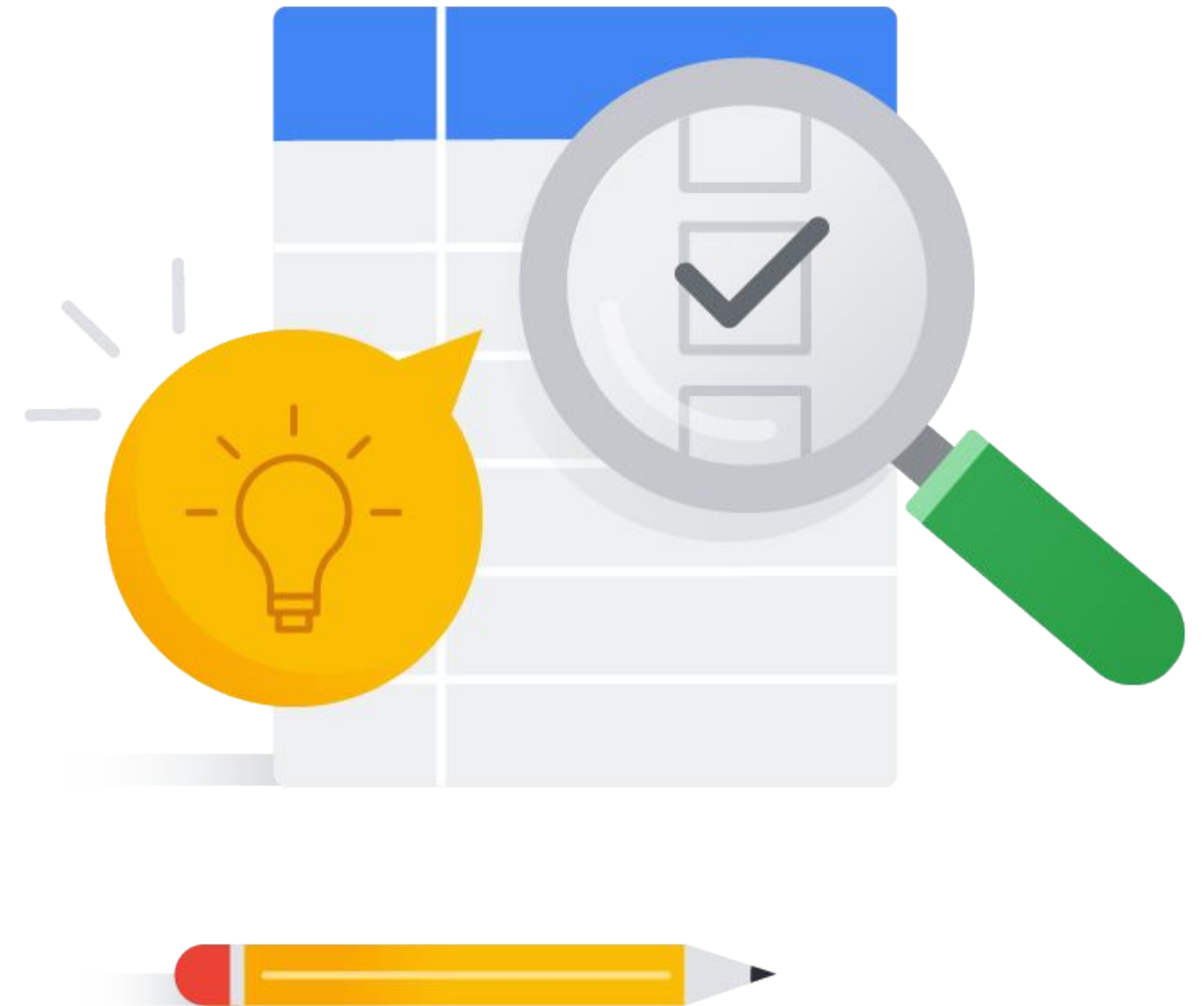
vertexai.init(project="982785856251", location="us-central1")
parameters = {
    "temperature": 0.2,
    "max_output_tokens": 256,
    "top_p": 0.8,
    "top_k": 40
}

model = TextGenerationModel.from_pretrained("text-bison@001")
model = model.get_tuned_model("projects/982785856251/locations/us-central1/models/167990643268360")
response = model.predict(
    """
    """,
    **parameters
)
print(f"Response from Model: {response.text}")
```

Demo

🕒 15 min ⚙️

Fine-Tuning Models for Specific Use Cases



Please mark your attendance at
goo.gle/genai-checkin

Thank you for attending this training!

We would love your feedback! Please take 3-5 minutes to complete our survey and help inform content and program related improvements.

- 1 Scan our QR Code
- 2 Enter the attendance code provided by your instructor
- 3 Complete the survey



In this module, you learned to ...

- 01 Evaluate scenarios for creating tuned models
- 02 Build workflows for tuning and deploying models
- 03 Use tuned models in your applications



