

# welcome-to-colab-1

July 27, 2025

## 1 Welcome to Colab!

### 1.1 Access Popular LLMs via Google-Colab-AI Without an API Key

Users with Colab's paid plans have free access to most popular LLMs via google-colab-ai Python library. For more details, refer to the [getting started with google colab ai](#).

```
from google.colab import ai
response = ai.generate_text("What is the capital of France?")
print(response)
```

### 1.2 Explore the Gemini API

The Gemini API gives you access to Gemini models created by Google DeepMind. Gemini models are built from the ground up to be multimodal, so you can reason seamlessly across text, images, code, and audio.

**How to get started?** \* Go to [Google AI Studio](#) and log in with your Google account. \* [Create an API key](#). \* Use a quickstart for [Python](#), or call the REST API using [curl](#).

**Discover Gemini's advanced capabilities** \* Play with Gemini [multimodal outputs](#), mixing text and images in an iterative way. \* Discover the [multimodal Live API](#) (demo [here](#)). \* Learn how to [analyze images and detect items in your pictures](#) using Gemini (bonus, there's a [3D version](#) as well!). \* Unlock the power of [Gemini thinking model](#), capable of solving complex task with its inner thoughts.

**Explore complex use cases** \* Use [Gemini grounding capabilities](#) to create a report on a company based on what the model can find on internet. \* Extract [invoices and form data from PDF](#) in a structured way. \* Create [illustrations based on a whole book](#) using Gemini large context window and Imagen.

To learn more, check out the [Gemini cookbook](#) or visit the [Gemini API documentation](#).

Colab now has AI features powered by [Gemini](#). The video below provides information on how to use these features, whether you're new to Python, or a seasoned veteran.

What is Colab?

Colab, or “Colaboratory”, allows you to write and execute Python in your browser, with - Zero configuration required - Access to GPUs free of charge - Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) or [Colab Features You May Have Missed](#) to learn more, or

just get started below!

### 1.3 Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
[ ]: seconds_in_a_day = 24 * 60 * 60
      seconds_in_a_day
```

```
[ ]: 86400
```

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut “Command/Ctrl+Enter”. To edit the code, just click the cell and start editing.

Variables that you define in one cell can later be used in other cells:

```
[ ]: seconds_in_a_week = 7 * seconds_in_a_day
      seconds_in_a_week
```

```
[ ]: 604800
```

Colab notebooks allow you to combine **executable code** and **rich text** in a single document, along with **images**, **HTML**, **LaTeX** and more. When you create your own Colab notebooks, they are stored in your Google Drive account. You can easily share your Colab notebooks with co-workers or friends, allowing them to comment on your notebooks or even edit them. To learn more, see [Overview of Colab](#). To create a new Colab notebook you can use the File menu above, or use the following link: [create a new Colab notebook](#).

Colab notebooks are Jupyter notebooks that are hosted by Colab. To learn more about the Jupyter project, see [jupyter.org](#).

### 1.4 Data science

With Colab you can harness the full power of popular Python libraries to analyze and visualize data. The code cell below uses **numpy** to generate some random data, and uses **matplotlib** to visualize it. To edit the code, just click the cell and start editing.

You can import your own data into Colab notebooks from your Google Drive account, including from spreadsheets, as well as from Github and many other sources. To learn more about importing data, and how Colab can be used for data science, see the links below under Working with Data.

```
[ ]: import numpy as np
      import IPython.display as display
      from matplotlib import pyplot as plt
      import io
      import base64
```

```

ys = 200 + np.random.randn(100)
x = [x for x in range(len(ys))]

fig = plt.figure(figsize=(4, 3), facecolor='w')
plt.plot(x, ys, '-')
plt.fill_between(x, ys, 195, where=(ys > 195), facecolor='g', alpha=0.6)
plt.title("Sample Visualization", fontsize=10)

data = io.BytesIO()
plt.savefig(data)
image = F"data:image/png;base64,{base64.b64encode(data.getvalue()).decode()}"
alt = "Sample Visualization"
display.display(display.Markdown(F"!!! [{alt}] ({image}) """))
plt.close(fig)

```

Colab notebooks execute code on Google's cloud servers, meaning you can leverage the power of Google hardware, including GPUs and TPUs, regardless of the power of your machine. All you need is a browser.

For example, if you find yourself waiting for **pandas** code to finish running and want to go faster, you can switch to a GPU Runtime and use libraries like [RAPIDS cuDF](#) that provide zero-code-change acceleration.

To learn more about accelerating pandas on Colab, see the [10 minute guide](#) or [US stock market data analysis demo](#).

## 1.5 Machine learning

With Colab you can import an image dataset, train an image classifier on it, and evaluate the model, all in just [a few lines of code](#).

Colab is used extensively in the machine learning community with applications including: - Getting started with TensorFlow - Developing and training neural networks - Experimenting with TPUs - Disseminating AI research - Creating tutorials

To see sample Colab notebooks that demonstrate machine learning applications, see the machine learning examples below.

## 1.6 More Resources

### 1.6.1 Working with Notebooks in Colab

- [Overview of Colab](#)
- [Guide to Markdown](#)
- [Importing libraries and installing dependencies](#)
- [Saving and loading notebooks in GitHub](#)
- [Interactive forms](#)

- [Interactive widgets](#)

### ### Working with Data

- [Loading data: Drive, Sheets, and Google Cloud Storage](#)
- [Charts: visualizing data](#)
- [Getting started with BigQuery](#)

## 1.6.2 Machine Learning

These are a few of the notebooks related to Machine Learning, including Google's online Machine Learning course. See the [full course website](#) for more. - [Intro to Pandas DataFrame](#) - [Intro to RAPIDS cuDF to accelerate pandas](#) - [Getting Started with cuML's accelerator mode](#) - [Linear regression with tf.keras using synthetic data](#)

### ### Using Accelerated Hardware

- [TensorFlow with GPUs](#)
- [TPUs in Colab](#)

## 1.6.3 Featured examples

- [Retraining an Image Classifier](#): Build a Keras model on top of a pre-trained image classifier to distinguish flowers.
- [Text Classification](#): Classify IMDB movie reviews as either *positive* or *negative*.
- [Style Transfer](#): Use deep learning to transfer style between images.
- [Multilingual Universal Sentence Encoder Q&A](#): Use a machine learning model to answer questions from the SQuAD dataset.
- [Video Interpolation](#): Predict what happened in a video between the first and the last frame.

```
[ ]: 
```

```
[ ]: 
```

```
[ ]: 
```

```
[ ]: 
```

```
[ ]: 
```

```
[1]: from google.colab import drive
      drive.mount('/content/drive')
```

Mounted at /content/drive

```
[ ]: 
```

```
[ ]: 
```

```
[2]: !nvidia-smi
```

Sun Jul 27 05:39:21 2025

```
+-----+
-----+
| NVIDIA-SMI 550.54.15              Driver Version: 550.54.15      CUDA Version:
12.4          |
|-----+-----+-----+
-----+
| GPU  Name                Persistence-M | Bus-Id        Disp.A | Volatile
Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |      Memory-Usage | GPU-Util
Compute M. |
|                |                  |
MIG M. |
|=====+=====+=====+
=====|
|   0   Tesla T4                       Off |  00000000:00:04.0 Off |
0 |
| N/A    42C    P8              10W /   70W |      0MiB /  15360MiB |      0%
Default |
|                |                  |
N/A |
+-----+-----+-----+
-----+

+-----+
-----+
| Processes:
|
| GPU   GI    CI          PID    Type    Process name
GPU Memory |
|       ID    ID
Usage      |
|=====+=====+=====+
=====|
| No running processes found
|
+-----+
-----+
```

```
[ ]: from diffusers import DiffusionPipeline
import torch

pipe = DiffusionPipeline.from_pretrained("stabilityai/
↳stable-diffusion-xl-base-1.0", torch_dtype=torch.float16,
↳use_safetensors=True, variant="fp16")
pipe.to("cuda")
```

```
# if using torch < 2.0
# pipe.enable_xformers_memory_efficient_attention()

prompt = "An astronaut riding a green horse"

images = pipe(prompt=prompt).images[0]
```

```
[18]: from transformers import pipeline
generator=pipeline("Text_generator",model='gpt2')
response=generator('what is huuging face',max_length=40,num_return_sequences=1)
print(response[0]['generated_text'])
```

```
-----
KeyError                                Traceback (most recent call last)
/tmp/ipython-input-18-1893231641.py in <cell line: 0>()
      1 from transformers import pipeline
----> 2 generator=pipeline("Text_generator",model='gpt2')
      3 response=generator('what is huuging
↳face',max_length=40,num_return_sequences=1)
      4 print(response[0]['generated_text'])

/usr/local/lib/python3.11/dist-packages/transformers/pipelines/__init__.py in
↳pipeline(task, model, config, tokenizer, feature_extractor, image_processor,
↳processor, framework, revision, use_fast, token, device, device_map,
↳torch_dtype, trust_remote_code, model_kwargs, pipeline_class, **kwargs)
    973         )
    974     else:
--> 975         normalized_task, targeted_task, task_options = check_task(task)
    976         if pipeline_class is None:
    977             pipeline_class = targeted_task["impl"]

/usr/local/lib/python3.11/dist-packages/transformers/pipelines/__init__.py in
↳check_task(task)
    546
    547     """
--> 548     return PIPELINE_REGISTRY.check_task(task)
    549
    550

/usr/local/lib/python3.11/dist-packages/transformers/pipelines/base.py in
↳check_task(self, task)
    1544         raise KeyError(f"Invalid translation task {task}, use
↳'translation_XX_to_YY' format")
    1545
-> 1546         raise KeyError(
    1547             f"Unknown task {task}, available tasks are {self.
↳get_supported_tasks() + ['translation_XX_to_YY']}"
```

1548 )

```
KeyError: "Unknown task Text_generator, available tasks are
↳['audio-classification', 'automatic-speech-recognition', 'depth-estimation',
↳'document-question-answering', 'feature-extraction', 'fill-mask',
↳'image-classification', 'image-feature-extraction', 'image-segmentation',
↳'image-text-to-text', 'image-to-image', 'image-to-text', 'mask-generation',
↳'ner', 'object-detection', 'question-answering', 'sentiment-analysis',
↳'summarization', 'table-question-answering', 'text-classification',
↳'text-generation', 'text-to-audio', 'text-to-speech', 'text2text-generation',
↳'token-classification', 'translation', 'video-classification',
↳'visual-question-answering', 'vqa', 'zero-shot-audio-classification',
↳'zero-shot-classification', 'zero-shot-image-classification',
↳'zero-shot-object-detection', 'translation_XX_to_YY']"
```

[ ]:

[ ]:

[ ]:

```
[17]: from transformers import pipeline
ner=pipeline('ner',grouped_entities=True)
entities=ner("hugging face is based on NYC and partner with google with $200_
↳dollar price")
print(entities)
```

No model was supplied, defaulted to dbmdz/bert-large-cased-finetuned-conll03-english and revision 4c53496 (<https://huggingface.co/dbmdz/bert-large-cased-finetuned-conll03-english>).

Using a pipeline without specifying a model name and revision in production is not recommended.

Some weights of the model checkpoint at dbmdz/bert-large-cased-finetuned-conll03-english were not used when initializing BertForTokenClassification:

['bert.pooler.dense.bias', 'bert.pooler.dense.weight']

- This IS expected if you are initializing BertForTokenClassification from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a BertForSequenceClassification model from a BertForPreTraining model).

- This IS NOT expected if you are initializing BertForTokenClassification from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceClassification model from a BertForSequenceClassification model).

Device set to use cuda:0

```
[{'entity_group': 'LOC', 'score': np.float32(0.99473166), 'word': 'NYC',
'start': 26, 'end': 29}, {'entity_group': 'ORG', 'score': np.float32(0.6143996),
'word': '##le', 'start': 51, 'end': 53}]
```

[ ]:

```
[ ]: 
[ ]: 
[ ]: 
[ ]: 
[19]: !pip install transformers datasets
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.11/dist-packages (4.53.3)
Requirement already satisfied: datasets in /usr/local/lib/python3.11/dist-packages (4.0.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from transformers) (3.18.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.30.0 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.33.4)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-packages (from transformers) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from transformers) (25.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages (from transformers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.11/dist-packages (from transformers) (2024.11.6)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from transformers) (2.32.3)
Requirement already satisfied: tokenizers<0.22,>=0.21 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.21.2)
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.5.3)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.11/dist-packages (from transformers) (4.67.1)
Requirement already satisfied: pyarrow>=15.0.0 in /usr/local/lib/python3.11/dist-packages (from datasets) (18.1.0)
Requirement already satisfied: dill<0.3.9,>=0.3.0 in /usr/local/lib/python3.11/dist-packages (from datasets) (0.3.8)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (from datasets) (2.2.2)
Requirement already satisfied: xxhash in /usr/local/lib/python3.11/dist-packages (from datasets) (3.5.0)
Requirement already satisfied: multiprocessing<0.70.17 in /usr/local/lib/python3.11/dist-packages (from datasets) (0.70.16)
Requirement already satisfied: fsspec<=2025.3.0,>=2023.1.0 in /usr/local/lib/python3.11/dist-packages (from fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (2025.3.0)
```



Requirement already satisfied: aiohttp!=4.0.0a0,!4.0.0a1 in /usr/local/lib/python3.11/dist-packages (from fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (3.12.14)

Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.30.0->transformers) (4.14.1)

Requirement already satisfied: hf-xet<2.0.0,>=1.1.2 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.30.0->transformers) (1.1.5)

Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (3.4.2)

Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (3.10)

Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (2.5.0)

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (2025.7.14)

Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas->datasets) (2.9.0.post0)

Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas->datasets) (2025.2)

Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas->datasets) (2025.2)

Requirement already satisfied: aiohappyeyeballs>=2.5.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp!=4.0.0a0,!4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (2.6.1)

Requirement already satisfied: aiosignal>=1.4.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp!=4.0.0a0,!4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (1.4.0)

Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp!=4.0.0a0,!4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (25.3.0)

Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.11/dist-packages (from aiohttp!=4.0.0a0,!4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (1.7.0)

Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.11/dist-packages (from aiohttp!=4.0.0a0,!4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (6.6.3)

Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp!=4.0.0a0,!4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (0.3.2)

Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp!=4.0.0a0,!4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (1.20.1)

Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-

packages (from python-dateutil>=2.8.2->pandas->datasets) (1.17.0)

```
[22]: from transformers import pipeline
classifier=pipeline('sentiment-analysis')
result=classifier('I love hugging face library ver much')
print(result)
```

No model was supplied, defaulted to distilbert/distilbert-base-uncased-finetuned-sst-2-english and revision 714eb0f  
(<https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english>).

Using a pipeline without specifying a model name and revision in production is not recommended.

Device set to use cuda:0

```
[{'label': 'POSITIVE', 'score': 0.999457061290741}]
```

```
[21]: from transformers import pipeline
classifier=pipeline('sentiment-analysis')
result=classifier('foof is not tasty')
print(result)
```

No model was supplied, defaulted to distilbert/distilbert-base-uncased-finetuned-sst-2-english and revision 714eb0f  
(<https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english>).

Using a pipeline without specifying a model name and revision in production is not recommended.

Device set to use cuda:0

```
[{'label': 'NEGATIVE', 'score': 0.999782383441925}]
```

```
[23]: !pip install gradio
```

Requirement already satisfied: gradio in /usr/local/lib/python3.11/dist-packages (5.38.1)

Requirement already satisfied: aiofiles<25.0,>=22.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (24.1.0)

Requirement already satisfied: anyio<5.0,>=3.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (4.9.0)

Requirement already satisfied: brotli>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (1.1.0)

Requirement already satisfied: fastapi<1.0,>=0.115.2 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.116.1)

Requirement already satisfied: ffmpy in /usr/local/lib/python3.11/dist-packages (from gradio) (0.6.1)

Requirement already satisfied: gradio-client==1.11.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (1.11.0)

Requirement already satisfied: groovy~=0.1 in /usr/local/lib/python3.11/dist-

packages (from gradio) (0.1.2)  
 Requirement already satisfied: httpx<1.0,>=0.24.1 in  
 /usr/local/lib/python3.11/dist-packages (from gradio) (0.28.1)  
 Requirement already satisfied: huggingface-hub>=0.28.1 in  
 /usr/local/lib/python3.11/dist-packages (from gradio) (0.33.4)  
 Requirement already satisfied: jinja2<4.0 in /usr/local/lib/python3.11/dist-  
 packages (from gradio) (3.1.6)  
 Requirement already satisfied: markupsafe<4.0,>=2.0 in  
 /usr/local/lib/python3.11/dist-packages (from gradio) (3.0.2)  
 Requirement already satisfied: numpy<3.0,>=1.0 in  
 /usr/local/lib/python3.11/dist-packages (from gradio) (2.0.2)  
 Requirement already satisfied: orjson~=3.0 in /usr/local/lib/python3.11/dist-  
 packages (from gradio) (3.11.0)  
 Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-  
 packages (from gradio) (25.0)  
 Requirement already satisfied: pandas<3.0,>=1.0 in  
 /usr/local/lib/python3.11/dist-packages (from gradio) (2.2.2)  
 Requirement already satisfied: pillow<12.0,>=8.0 in  
 /usr/local/lib/python3.11/dist-packages (from gradio) (11.3.0)  
 Requirement already satisfied: pydantic<2.12,>=2.0 in  
 /usr/local/lib/python3.11/dist-packages (from gradio) (2.11.7)  
 Requirement already satisfied: pydub in /usr/local/lib/python3.11/dist-packages  
 (from gradio) (0.25.1)  
 Requirement already satisfied: python-multipart>=0.0.18 in  
 /usr/local/lib/python3.11/dist-packages (from gradio) (0.0.20)  
 Requirement already satisfied: pyyaml<7.0,>=5.0 in  
 /usr/local/lib/python3.11/dist-packages (from gradio) (6.0.2)  
 Requirement already satisfied: ruff>=0.9.3 in /usr/local/lib/python3.11/dist-  
 packages (from gradio) (0.12.4)  
 Requirement already satisfied: safehttpx<0.2.0,>=0.1.6 in  
 /usr/local/lib/python3.11/dist-packages (from gradio) (0.1.6)  
 Requirement already satisfied: semantic-version~=2.0 in  
 /usr/local/lib/python3.11/dist-packages (from gradio) (2.10.0)  
 Requirement already satisfied: starlette<1.0,>=0.40.0 in  
 /usr/local/lib/python3.11/dist-packages (from gradio) (0.47.2)  
 Requirement already satisfied: tomlkit<0.14.0,>=0.12.0 in  
 /usr/local/lib/python3.11/dist-packages (from gradio) (0.13.3)  
 Requirement already satisfied: typer<1.0,>=0.12 in  
 /usr/local/lib/python3.11/dist-packages (from gradio) (0.16.0)  
 Requirement already satisfied: typing-extensions~=4.0 in  
 /usr/local/lib/python3.11/dist-packages (from gradio) (4.14.1)  
 Requirement already satisfied: uvicorn>=0.14.0 in  
 /usr/local/lib/python3.11/dist-packages (from gradio) (0.35.0)  
 Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages  
 (from gradio-client==1.11.0->gradio) (2025.3.0)  
 Requirement already satisfied: websockets<16.0,>=10.0 in  
 /usr/local/lib/python3.11/dist-packages (from gradio-client==1.11.0->gradio)  
 (15.0.1)

Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.11/dist-packages (from anyio<5.0,>=3.0->gradio) (3.10)

Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.11/dist-packages (from anyio<5.0,>=3.0->gradio) (1.3.1)

Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-packages (from httpx<1.0,>=0.24.1->gradio) (2025.7.14)

Requirement already satisfied: httpcore==1.\* in /usr/local/lib/python3.11/dist-packages (from httpx<1.0,>=0.24.1->gradio) (1.0.9)

Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.11/dist-packages (from httpcore==1.\*->httpx<1.0,>=0.24.1->gradio) (0.16.0)

Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradio) (3.18.0)

Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradio) (2.32.3)

Requirement already satisfied: tqdm==4.42.1 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradio) (4.67.1)

Requirement already satisfied: hf-xet<2.0.0,>=1.1.2 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradio) (1.1.5)

Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2.9.0.post0)

Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2025.2)

Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2025.2)

Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradio) (0.7.0)

Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradio) (2.33.2)

Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradio) (0.4.1)

Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0,>=0.12->gradio) (8.2.1)

Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0,>=0.12->gradio) (1.5.4)

Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0,>=0.12->gradio) (13.9.4)

Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas<3.0,>=1.0->gradio) (1.17.0)

Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from rich>=10.11.0->typer<1.0,>=0.12->gradio) (3.0.0)

Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from

```
rich>=10.11.0->typer<1.0,>=0.12->gradio) (2.19.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests->huggingface-
hub>=0.28.1->gradio) (3.4.2)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.11/dist-packages (from requests->huggingface-
hub>=0.28.1->gradio) (2.5.0)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-
packages (from markdown-it-py>=2.2.0->rich>=10.11.0->typer<1.0,>=0.12->gradio)
(0.1.2)
```

```
[2]: import gradio as gr
      from transformers import pipeline
      classifier=pipeline("sentiment-analysis")
      def analyze_sentiment(text):
          return classifier(text)[0]['label']
      gr.Interface(fn=analyze_sentiment,inputs='text',outputs='text').launch()
```

No model was supplied, defaulted to distilbert/distilbert-base-uncased-  
finetuned-sst-2-english and revision 714eb0f  
(<https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english>).

Using a pipeline without specifying a model name and revision in production is  
not recommended.

```
config.json: 0%|          | 0.00/629 [00:00<?, ?B/s]
model.safetensors: 0%|          | 0.00/268M [00:00<?, ?B/s]
tokenizer_config.json: 0%|          | 0.00/48.0 [00:00<?, ?B/s]
vocab.txt: 0.00B [00:00, ?B/s]
Device set to use cuda:0
```

It looks like you are running Gradio on a hosted Jupyter notebook, which  
requires `share=True`. Automatically setting `share=True` (you can turn this off  
by setting `share=False` in `launch()` explicitly).

Colab notebook detected. To show errors in colab notebook, set debug=True in  
launch()

\* Running on public URL: <https://a84462696790ab732d.gradio.live>

This share link expires in 1 week. For free permanent hosting and GPU upgrades,  
run `gradio deploy` from the terminal in the working directory to deploy to  
Hugging Face Spaces (<https://huggingface.co/spaces>)

<IPython.core.display.HTML object>

[2]: