



Machine Learning with Hugging Face

Host: Ahad Jawaid

Who am I?



Ahad Jawaid

Computer Science Undergraduate Student, The University of Texas at Dallas

Advisor: Dr. Berrak Sisman

Research Field: emotional text-to-speech synthesis



What I expect from you

```
def add(a, b):  
    return a + b
```

```
class Point:  
    def __init__(self, x, y):  
        self.x = x  
        self.y = y  
  
    def calculate_hyptonuse(self):  
        return (self.x**2 + self.y**2)**0.5
```

Prerequisites



Gmail

or



Overview

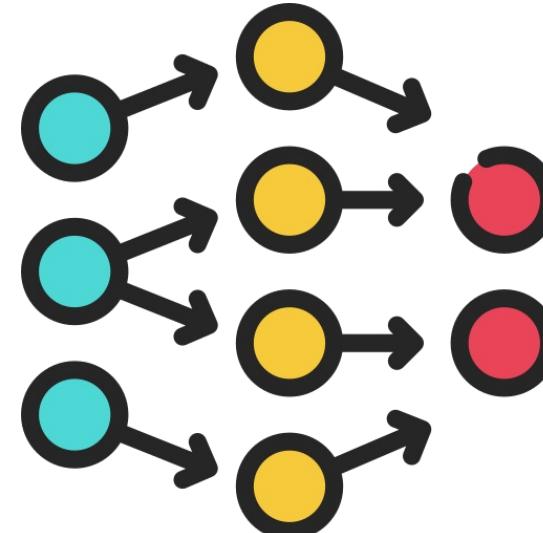
GitHub Link



github.com/ahadjawaid/hackutd-hf-workshop

Time	Section
5 minutes	Introduction
5 minutes	Overview of ML
5 minutes	Setup Hugging Face
5 minutes	Setup the environment
20 minutes	Using a Model
10 minutes	Using a Dataset
5 minutes	Hugging Face Spaces
5 minutes	Q&A

What is Machine Learning?



How it works

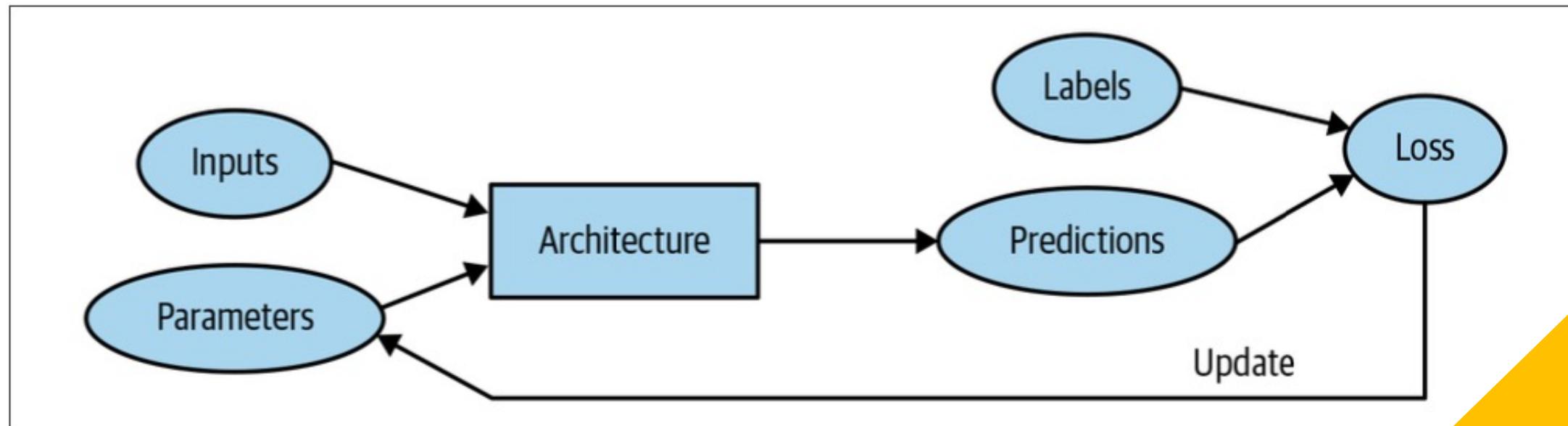
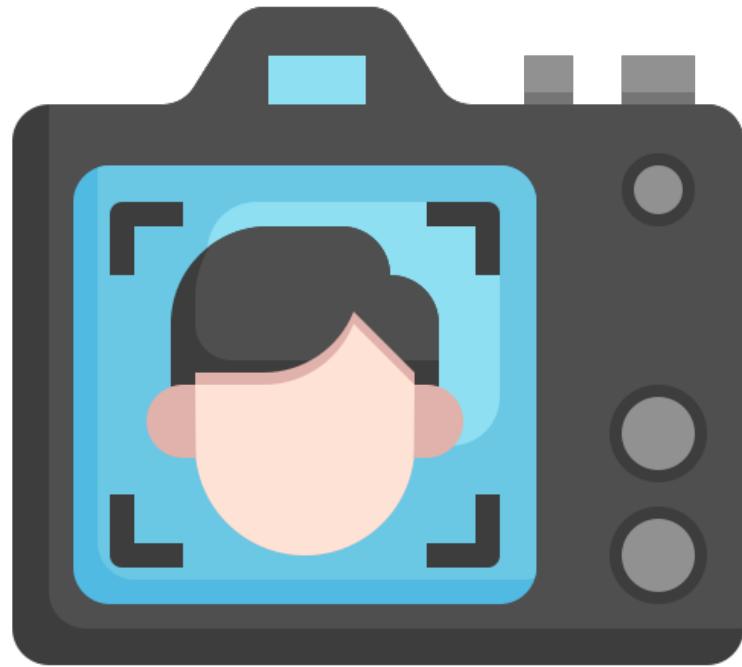


Figure 1-8. Detailed training loop

What are datasets?

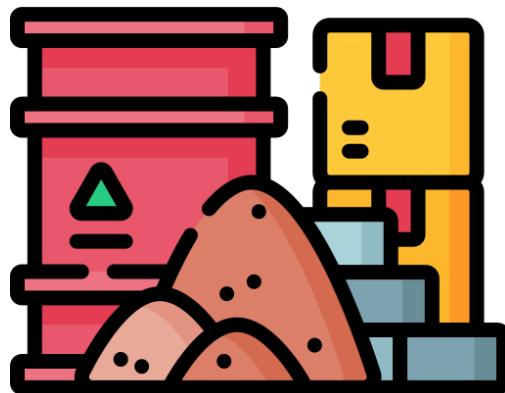


Inputs



Desired outputs

What is a pretrained model?



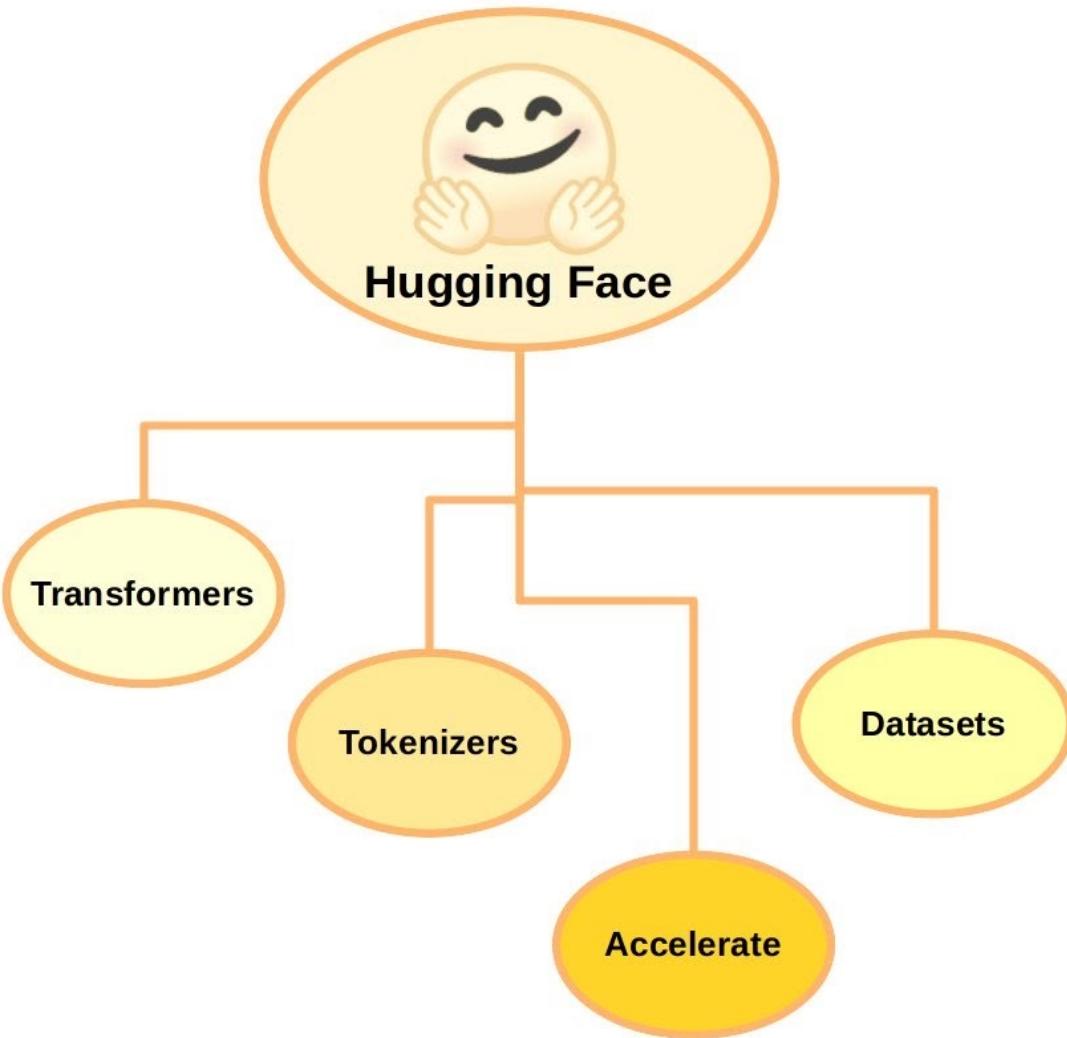
Untrained model



Pretrained model

What does deploying mean?





What is
Hugging
Face



Join Hugging Face

Join the community of machine learners!

Email Address

 ...

Hint: Use your organization email to easily find and join your company/team org.

Password

 ...

Next

Already have an account? [Log in](#)

SSO is available for companies



Complete your profile

One last step to join the community

Username ... Full name

Avatar (optional) GitHub username (optional) ...

Upload file GitHub username ...

Homepage (optional) Twitter username (optional)

Homepage Twitter account

Research interests (optional)

I have read and agree with the [Terms of Service](#) and the [Code of Conduct](#)

Create Account



Search models, datasets

≡

Please check your email address for a confirmation link

Resend confirmation email

Creating a Hugging Face (HF) account

Using Google Colab

The screenshot shows the Google Colab interface. At the top, there's a navigation bar with 'Help', 'Share', a gear icon, and a 'Sign in' button, which is highlighted with a red box and an arrow pointing to the right. Below the navigation bar, there are buttons for '+ Code' and '+ Text', and a 'Copy to Drive' button. The main content area starts with a 'Welcome to Colab!' message. Below it, a text block says: 'If you're already familiar with Colab, check out this video to learn about interactive tables, the executed code history view, and the command palette.' To the right of this text is a video thumbnail titled '3 Cool Google Colab Features' featuring a man speaking. At the bottom, there's a section titled 'What is Colab?' with the text: 'Colab, or "Colaboratory", allows you to write and execute Python in your browser, with'.

Sign in

Use your Google Account

Email or phone



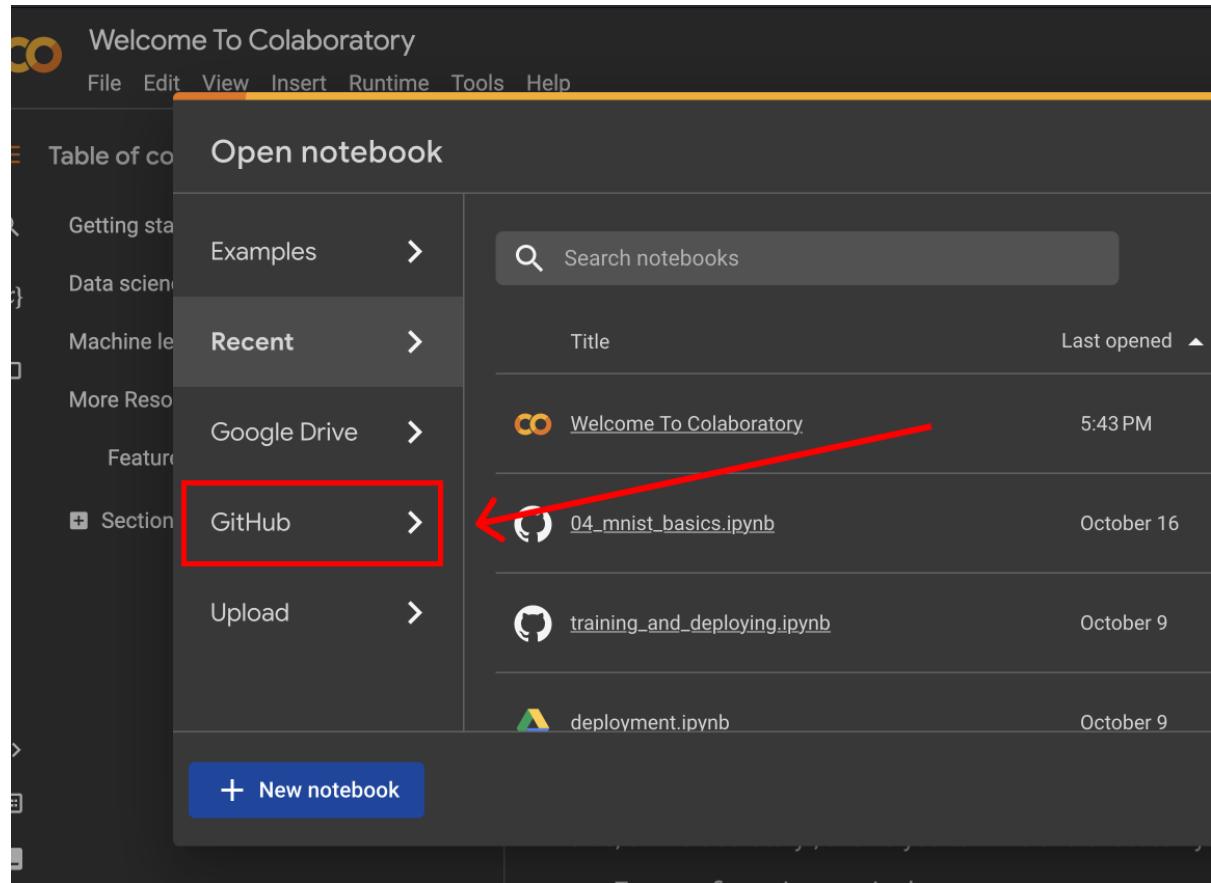
[Forgot email?](#)

Not your computer? Use a Private Window to sign in.
[Learn more](#)

[Create account](#)

[Next](#)

Using a notebook



Accessing hugging face

```
from huggingface_hub import notebook_login  
  
notebook_login()
```



Copy a token from [your Hugging Face tokens page](#) and paste it below.

Immediately click login after copying your token or it might be stored in plain text
in this notebook file.

Token:

Add token as git credential?

Login

Pro Tip: If you don't already have one, you can create a dedicated 'notebooks'
token with 'write' access, that you can then easily reuse for all notebooks.

Getting hugging face token

The image shows the Hugging Face user interface. At the top, there is a navigation bar with links: Models, Datasets, Spaces, Docs, Solutions, Pricing, and a user icon. A red arrow points from the user icon to a red-bordered box around it. Below the navigation bar, the user's profile is shown: Ahad Jawaid (ahad-j) with a fox icon. The profile menu includes options: Notifications (Inbox 0), New Model, New Dataset, New Space, New Collection, Create organization, Settings (which is highlighted with a red box and has a red arrow pointing to it), and Sign Out. To the right of the profile, a sidebar shows the user's account information: Ahad Jawaid (ahad-j) with a fox icon. The sidebar lists various sections: Profile, Account, Organizations, Billing, Access Tokens (which is highlighted with a red box and has a red arrow pointing to it), SSH and GPG Keys, Webhooks, Papers, Notifications, Content Preferences, Connected Apps, and Theme.

Models Datasets Spaces Docs Solutions Pricing

Profile
Ahad Jawaid
ahad-j

Notifications
Inbox (0)

New Model

New Dataset

New Space

New Collection

Create organization

Settings

Sign Out

Profile
Ahad Jawaid
ahad-j

Account

Organizations

Billing

Access Tokens

SSH and GPG Keys

Webhooks

Papers

Notifications

Content Preferences

Connected Apps

Theme

Getting hugging face token

Access Tokens

User Access Tokens

Access tokens programmatically authenticate your identity to the Hugging Face Hub, allowing applications to perform specific actions specified by the scope of permissions (read, write, or admin) granted. Visit [the documentation](#) to discover how to use them.

git WRITE
.....
Manage Show

git READ
.....
Manage Show

nb WRITE
.....
Manage Show

New token

Create a new access token

Name: HackUTD

Role: read

Generate a token

A red box highlights the 'Generate a token' button, and a red arrow points from the 'Name' field to it.

```
from huggingface_hub import notebook_login  
  
notebook_login()
```



Copy a token from your [Hugging Face tokens page](#) and paste it below.

Immediately click login after copying your token or it might be stored in plain text
in this notebook file.

Token:

Add token as git credential?

Login

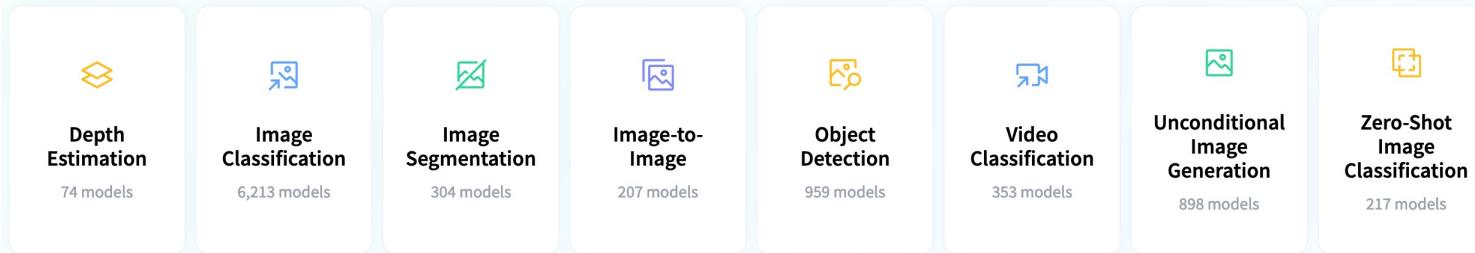
Pro Tip: If you don't already have one, you can create a dedicated 'notebooks'
token with 'write' access, that you can then easily reuse for all notebooks.

Accessing hugging face

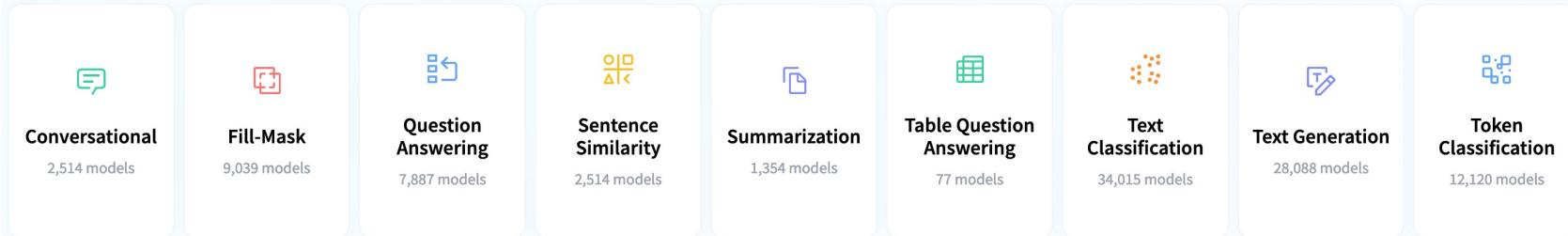
● Tasks

Hugging Face is the home for all Machine Learning tasks. Here you can find what you need to get started with a task: demos, use cases, models, datasets, and more!

Computer Vision



Natural Language Processing



Browsing Hugging Face Tasks

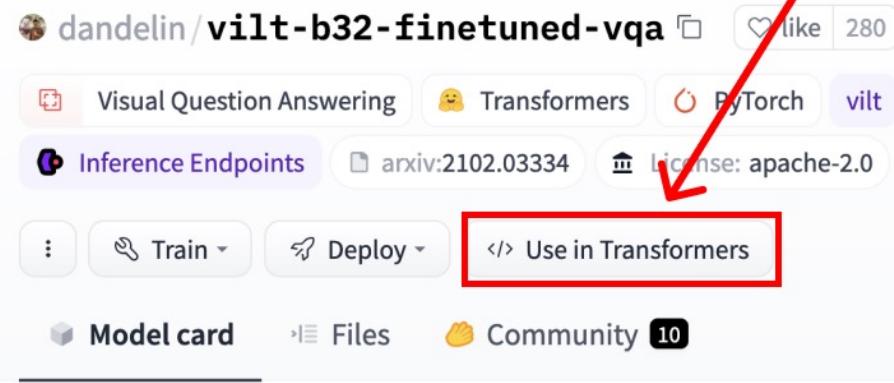
<https://huggingface.co/tasks>

The screenshot shows the Hugging Face Model Hub interface. At the top, there are navigation links: Tasks (1), Libraries, Datasets, Languages, Licenses, and Other. Below these are search and filter fields: 'Filter Tasks by name' and 'Reset Tasks'. The main content area is titled 'Models 1,659' with a 'Filter by name' dropdown. To the right are buttons for 'new', 'Full-text search', and 'Sort: Trending'. The page displays a list of models, each with a checkbox, the model name, its type, last update, and metrics (downloads and stars). The models listed are:

- coqui/XTTS-v1: Text-to-Speech. Updated 1 day ago. 14.8k downloads, 289 stars.
- suno/bark: Text-to-Speech. Updated 18 days ago. 46.5k downloads, 492 stars.
- microsoft/speecht5_tts: Text-to-Speech. Updated 17 days ago. 48.6k downloads, 290 stars.
- facebook/fastspeech2-en-ljspeech: Text-to-Speech. Updated Jan 28, 2022. 8.24k downloads, 220 stars.
- voxxer/speecht5_finetuned_commonvoice_ru_translit: Text-to-Speech. Updated Aug 25. 88 downloads, 2 stars.
- facebook/mms-tts-eng: Text-to-Speech. Updated Sep 6. 10.1k downloads, 18 stars.
- kingsznhone/Red-Alert-2-Full-Voice-Model: (highlighted with a blue dot)

Importing Model

Using the model



</> How to use from the • **Transformers** ⓘ library X

```
# Use a pipeline as a high-level helper
from transformers import pipeline

pipe = pipeline("visual-question-answering", model="dandelin/vilt-b32-finetuned-vqa")

# Load model directly
from transformers import AutoProcessor, ViltForVisualQuestionAnswering

processor = AutoProcessor.from_pretrained("dandelin/vilt-b32-finetuned-vqa")
model = ViltForVisualQuestionAnswering.from_pretrained("dandelin/vilt-b32-finetuned-vqa")
```

Copy

Copy

Quick Links

- Read model documentation
- Read docs on high-level-pipeline
- Read our learning resources



Input

Using Pipeline

```
# Use a pipeline as a high-level helper
from transformers import pipeline

pipe = pipeline("visual-question-answering", model="dandelin/vilt-b32-finetuned-vqa")

from PIL import Image

image_path = "./person.jpg"
image = Image.open(image_path)
text = "What color shirt is the person wearing?"

out = pipe(image, text)

out
[{'score': 0.9601157307624817, 'answer': 'blue'},
 {'score': 0.5611773729324341, 'answer': 'black'},
 {'score': 0.09129240363836288, 'answer': 'navy'},
 {'score': 0.02366543561220169, 'answer': 'gray'},
 {'score': 0.0042873346246778965, 'answer': 'purple'}]
```

Using Model

```
# Load model directly
from transformers import ViltProcessor, ViltForQuestionAnswering

processor = ViltProcessor.from_pretrained("danelin/vilt-b32-finetuned-vqa")
model = ViltForQuestionAnswering.from_pretrained("danelin/vilt-b32-finetuned-vqa")

from PIL import Image

image_path = "./person.jpg"
image = Image.open(image_path)
text = "What color shirt is the person wearing?"

encoding = processor(image, text, return_tensors="pt")

outputs = model(**encoding)

sorted_indexes = outputs.logits.argsort(dim=-1, descending=True)[0].tolist()

top 5 prediciton = list(map(lambda index: model.config.id2label[index], sorted_indexes))[:5]
```

Using Model

```
# Load model directly
from transformers import ViltProcessor, ViltForQuestionAnswering

processor = ViltProcessor.from_pretrained("dandelin/vilt-b32-finetuned-vqa")
model = ViltForQuestionAnswering.from_pretrained("dandelin/vilt-b32-finetuned-vqa")

from PIL import Image

image_path = "./person.jpg"
image = Image.open(image_path)
text = "What color shirt is the person wearing?"

encoding = processor(image, text, return_tensors="pt")

outputs = model(**encoding)

sorted_indexs = outputs.logits.argsort(dim=-1, descending=True)[0].tolist()

top 5 prediciton = list(map(lambda index: model.config.id2label[index], sorted_indexs))[:5]
```

How datasets look like

Dataset Viewer (First 5GB) i

Auto-converted

Split

train (11.3k rows) v

image
image

artist
class label

genre
class label



22 vincent-van-gogh

4 landscape



20 rembrandt

7 religious_painting



16 paul-cezanne

6 portrait



17 pierre-auguste-renoir

2 genre_painting



9 ivan-aivazovsky

10 Unknown Genre

< Previous

1

2

3

...

114

Next

Browsing Hugging Face Datasets

The screenshot shows the Hugging Face Datasets homepage. On the left, there's a sidebar with navigation links: Tasks, Sizes, Sub-tasks, Languages, Licenses, Other, Filter Tasks by name, Multimodal (Feature Extraction, Text-to-Image, Image-to-Text, Text-to-Video, Visual Question Answering, Graph Machine Learning), Computer Vision (Depth Estimation, Image Classification, Object Detection, Image Segmentation, Image-to-Image, Unconditional Image Generation, Video Classification, Zero-Shot Image Classification), and Natural Language Processing.

The main area displays a list of 71,844 datasets. At the top right are buttons for Full-text search and Sort: Trending. Below is a grid of dataset cards:

- open-web-math/open-web-math (Viewer, Updated 5 days ago, 576, 146)
- openbmb/UltraFeedback (Viewer, Updated 22 days ago, 856, 111)
- approximatelabs/tablib-v1-full (Viewer, Updated 9 days ago, 23, 49)
- stingning/ultrachat (Viewer, Updated 11 days ago, 3.57k, 220)
- EleutherAI/proof-pile-2 (Viewer, Updated 5 days ago, 163, 34)
- laion/dalle-3-dataset (Viewer, Updated 31 minutes ago, 1.09k, 140)
- fka/awesome-chatgpt-prompts (Viewer, Updated Mar 7, 1.53k, 3.61k)
- Open-Orca/OpenOrca (Preview, Updated 1 day ago, 11.3k, 819)
- THUDM/AgentInstruct (Viewer, Updated 3 days ago, 61, 16)
- ehartford/dolphin (Preview, Updated 27 days ago, 472, 210)
- QingyiSi/Alpaca-CoT (Viewer, Updated Sep 14, 481, 509)
- Open-Orca/SlimOrca (Viewer, Updated 11 days ago, 596, 27)
- Yukang/LongAlpaca-12k (Viewer, Updated 11 days ago, 596, 27)
- akjindal53244/Arithmo-Data (Viewer, Updated 11 days ago, 596, 27)

Importing the dataset

```
from datasets import load_dataset  
  
dataset = load_dataset("yelp_review_full")  
  
Found cached dataset yelp_review_full (/Users/tk541/.cache/huggingface/datasets/yelp_review_full/yelp_review_full/1.0.0/e8e18e19d7be9e75642fc66b198ababd116f73599ec89a69ba5dd8d1e57ba0bf)  
  
100% [2/2 [00:00<00:00, 9.97it/s]
```

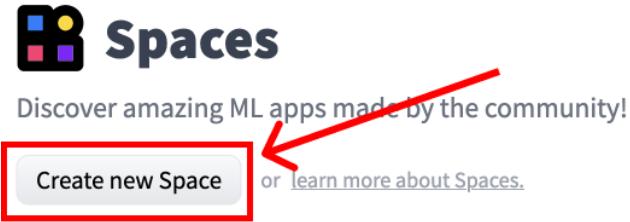
Viewing the Dataset

```
dataset  
  
DatasetDict({  
    train: Dataset({  
        features: ['label', 'text'],  
        num_rows: 650000  
    })  
    test: Dataset({  
        features: ['label', 'text'],  
        num_rows: 50000  
    })  
})
```

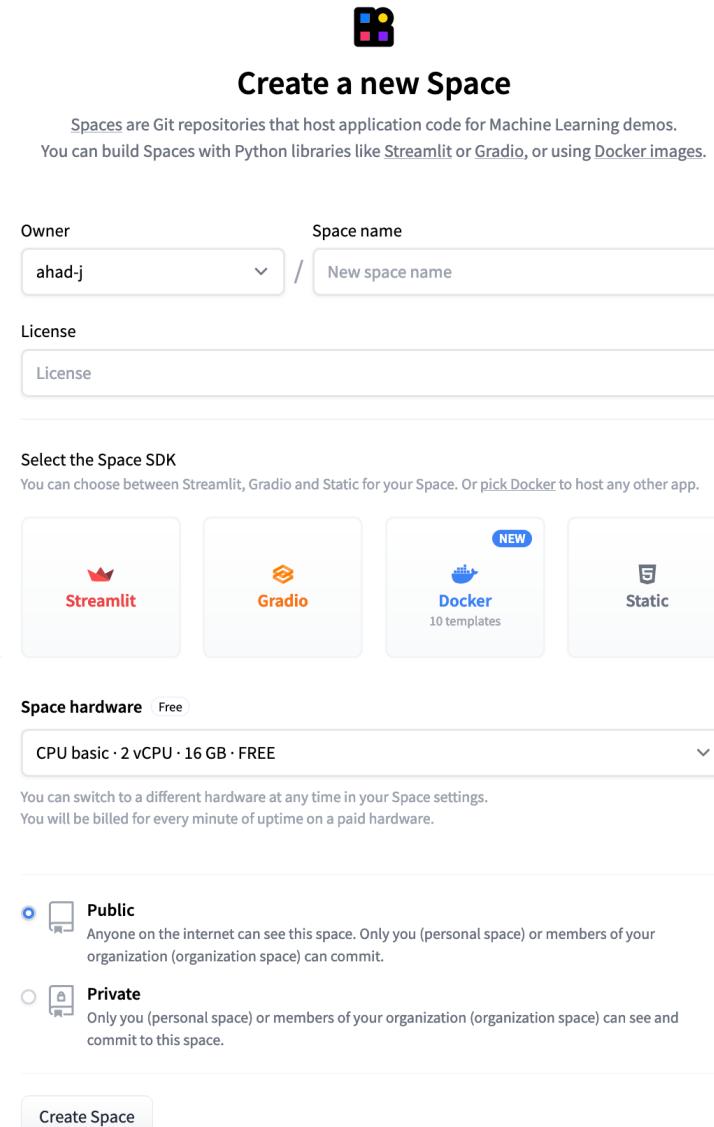
The Dataset is structured as a dictionary containing both the training, and test set, so to view a sample we must select which set we want to view from.

```
dataset["train"][0]  
  
{'label': 4,  
 'text': "dr. goldberg offers everything i look for in a general practitioner. he's nice and easy to talk to without being patronizing; he's always on time in seeing his patients; he's affiliated with a top-notch hospital (nyu) which my parents have explained to me is very important in case something happens and you need surgery; and you can get referrals to see specialists without having to see him first. really, what more do you need? i'm sitting here trying to think of any complaints i have about him, but i'm really drawing a blank."}
```

What is hugging face spaces



Spaces
Discover amazing ML apps made by the community!
[Create new Space](#) or [learn more about Spaces.](#)



Create a new Space

Spaces are Git repositories that host application code for Machine Learning demos. You can build Spaces with Python libraries like Streamlit or Gradio, or using Docker images.

Owner: ahad-j / Space name:

License:

Select the Space SDK: You can choose between Streamlit, Gradio and Static for your Space. Or pick Docker to host any other app.

Streamlit **Gradio** **Docker** (NEW) **Static**

Space hardware: CPU basic · 2 vCPU · 16 GB · FREE
You can switch to a different hardware at any time in your Space settings. You will be billed for every minute of uptime on a paid hardware.

Public: Anyone on the internet can see this space. Only you (personal space) or members of your organization (organization space) can commit.
 Private: Only you (personal space) or members of your organization (organization space) can see and commit to this space.

[Create Space](#)

Deploying on Spaces

ahad-j/dog-v-cat like 0 • Running

App Files Community Settings

image



Dog 100%

Cat 0%

Clear Submit

The screenshot shows a deployment interface for an application named "dog-v-cat". The top navigation bar includes the repository name "ahad-j/dog-v-cat", a "like" counter (0), a "Running" status indicator, and tabs for "App", "Files", "Community", and "Settings". The "App" tab is active. On the left, there's a file input field labeled "image" containing a photo of a golden retriever. Below it are "Clear" and "Submit" buttons. On the right, under the "output" section, the word "Dog" is prominently displayed above a progress bar at 100%, and "Cat" is shown at 0%.

Q&A

Feedback Form



<https://forms.gle/9gmj7wYjm1Rc5jhW6>