

Intro Hugging Face and Python

Dirk Wulff & Zak Hussain



MAX PLANCK INSTITUTE
FOR HUMAN DEVELOPMENT



OUR SOFTWARE STACK



+



+



OUR SOFTWARE STACK



+



+



WHY IS HUGGING FACE?



WHY IS HUGGING FACE?



Traditional language modelling pipeline:

WHY IS HUGGING FACE?



Traditional language modelling pipeline:

1. Find out the model architecture

WHY IS HUGGING FACE?

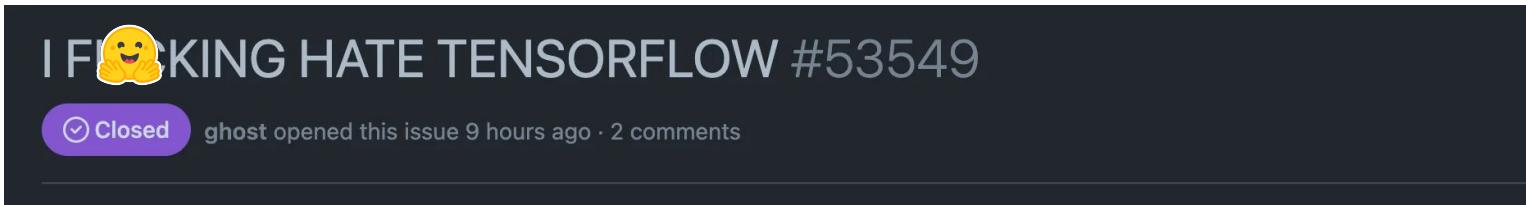


Traditional language modelling pipeline:

1. Find out the model architecture
2. Implement the model architecture in code with deep learning frameworks (e.g PyTorch/Tensorflow).

1. DEEP LEARNING LIBRARIES CAN BE DIFFICULT

1. DEEP LEARNING LIBRARIES CAN BE DIFFICULT



WHY HUGGING FACE?

Traditional language modelling pipeline:

1. Find out the model architecture
2. Implement the model architecture in code with deep learning libraries (e.g PyTorch/Tensorflow).

WHY HUGGING FACE?

Traditional language modelling pipeline:

1. Find out the model architecture
2. Implement the model architecture in code with deep learning libraries (e.g PyTorch/Tensorflow).
3. Load the pretrained weights (if available) from a server.
4. Process the inputs (using the correct tokenizer for the model)
5. Implement data loaders
6. Define a loss function
7. Stick a task-specific “head” on the model

HUGGING FACE PIPELINES

Import pipeline

```
from transformers import pipeline
```

Initialise
pipeline

```
pipe = pipeline('text-generation', model='gpt2')
```

Load model
input

```
prompt = """  
Once upon a time in a land far far away, there was a young prince named  
John. He was known for his bravery and courage. One day, he decided to go on  
an adventure to explore the unknown lands.  
"""
```

Feed input
the model

```
output = pipe(prompt, max_length=100)  
print(output)
```

HUGGING FACE

The screenshot shows the Hugging Face homepage. At the top, there is a search bar with placeholder text "Search models, datasets, users...". Below the search bar, there are navigation links for Models, Datasets, Spaces, Community, Docs, Enterprise, and Pricing. A user profile icon is also present.

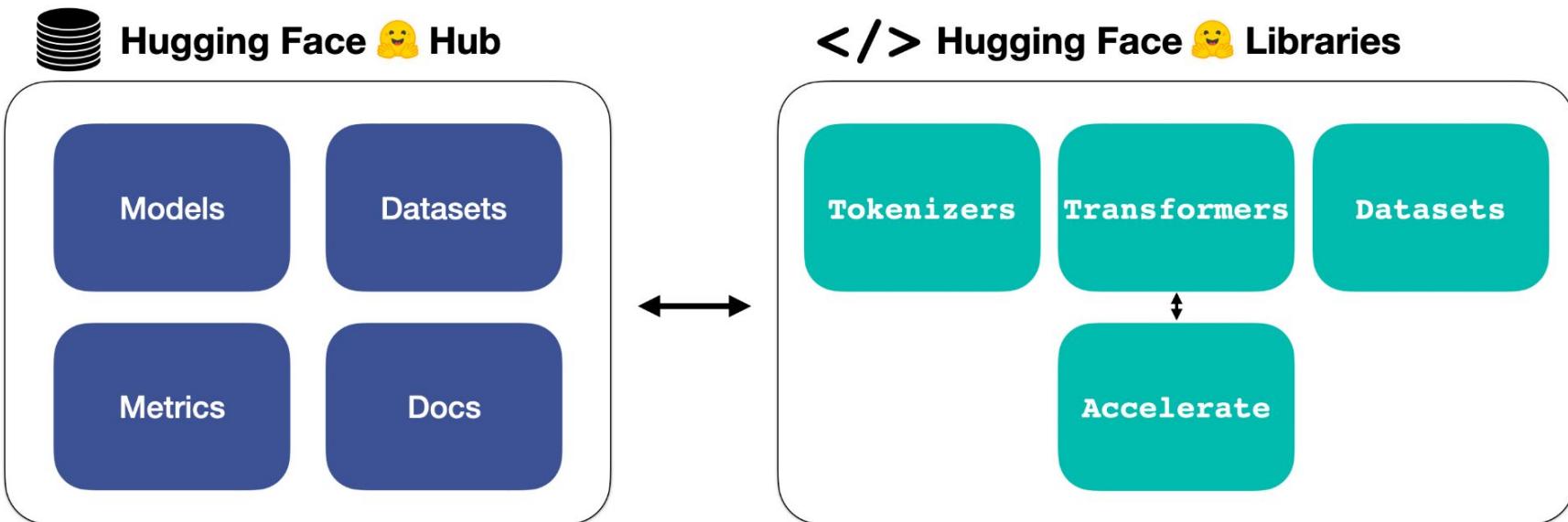
The main content area is divided into several sections:

- Main**: Includes links for Tasks, Libraries, Languages, Licenses, and Other. Under Tasks, there are categories for Text Generation, Any-to-Any, Image-Text-to-Text, Image-to-Text, Image-to-Image, Text-to-Image, Text-to-Video, and Text-to-Speech, with a total count of +42.
- Parameters**: A slider for adjusting model parameters, ranging from 1B to >500B.
- Libraries**: Categories for PyTorch, TensorFlow, JAX, Transformers, Diffusers, sentence-transformers, Safetensors, ONNX, GGU, Transformers.js, MLX, and +41.
- Apps**: Categories for vLLM, TGI, llama.cpp, MLX LM, LM Studio, Ollama, Jan, and +13.
- Inference Providers**: Categories for Groq, Novita, Nebulus AI, Cerebras, SambaNova, Nscale, fal, Hyperbolic, and +11.

The central part of the page displays a list of 2,256,270 models, filtered by name. The results are sorted by Trending. Each model entry includes the owner's profile picture, the model name, its type, last update time, size, and the number of stars and forks.

Model	Type	Last Update	Size	Stars	Forks
Tongyi-MAI/Z-Image-Turbo	Text-to-Image	Updated about 23 hours ago	~ 111k	1.91k	19
deepseek-ai/DeepSeek-V3.2	Text Generation	Updated 2 days ago	~ 5.01k	626	10
deepseek-ai/DeepSeek-V3.2-Speciale	Text Generation	Updated 2 days ago	~ 1.87k	439	10
black-forest-labs/FLUX.2-dev	Image-to-Image	Updated 6 days ago	~ 185k	850	10
Comfy-Org/z_image_turbo	Image-to-Image	Updated 6 days ago	~ 1.5M	339	10
tencent/HunyuanOCR	Image-Text-to-Text	~ 1.0B	~ 226k	614	10
microsoft/Fara-7B	Image-Text-to-Text	~ 8B	~ 25.3k	388	10
facebook/sam3	Mask Generation	~ 0.9B	~ 327k	858	10
apple/starflow	Image-to-Image	Updated 1 day ago	~ 172	172	10
PrimeIntellect/INTELLECT-3	Text Generation	~ 107B	~ 2.75k	170	10
alibaba-pai/Z-Image-Turbo-Fun-Controlnet-Union	Image-to-Image	Updated 1 day ago	~ 177	177	10
jayn7/Z-Image-Turbo-GGUF	Text-to-Image	~ 6B	~ 70k	154	10
AIDC-AI/Ovis-Image-7B	Text-to-Image	Updated about 8 hours ago	~ 1.26k	139	10
mistralai/Mistral-Large-3-675B-Instruct-2512	Text Generation	Updated about 19 hours ago	~ 112	113	10
stepfun-ai/Step-Audio-R1	Audio-Text-to-Text	~ 33B	~ 315	110	10
T5B/Z-Image-Turbo-FP8	Text-to-Image	Updated 6 days ago	~ 110k	91	10
salakash/SamKash-Tolstoy	Text Generation	Updated 6 days ago	~ 7.88k	469	10
arcee-ai/Trinity-Mini	Text Generation	Updated 1 day ago	~ 240	87	10
mistralai/Mistral-7-14B-Tortoise-2E12	Text Generation	Updated 1 day ago	~ 2E12	2E12	10

THE HUGGING FACE ECOSYSTEM



HUGGING FACE DOCUMENTATION

DOC Documentations

Search across all docs

Hub

Host Git-based models, datasets and Spaces on the Hugging Face Hub.

Hub Python Library

Client library for the HF Hub: manage repositories from your Python runtime.

Inference API

Use more than 50k models through our public inference API, with scalability built-in.

Transformers

State-of-the-art ML for Pytorch, TensorFlow, and JAX.

Datasets

Access and share datasets for computer vision, audio, and NLP tasks.

Huggingface.js

A collection of JS libraries to interact with Hugging Face, with TS types included.

Inference Endpoints

Easily deploy your model to production on dedicated, fully managed infrastructure.

Diffusers

State-of-the-art diffusion models for image and audio generation in PyTorch.

Gradio

Build machine learning demos and other web apps, in just a few lines of Python.

Transformers.js

Community library to run pretrained models from Transformers in your browser.

PEFT

Parameter efficient finetuning methods for large models

OUR SOFTWARE STACK



+



+





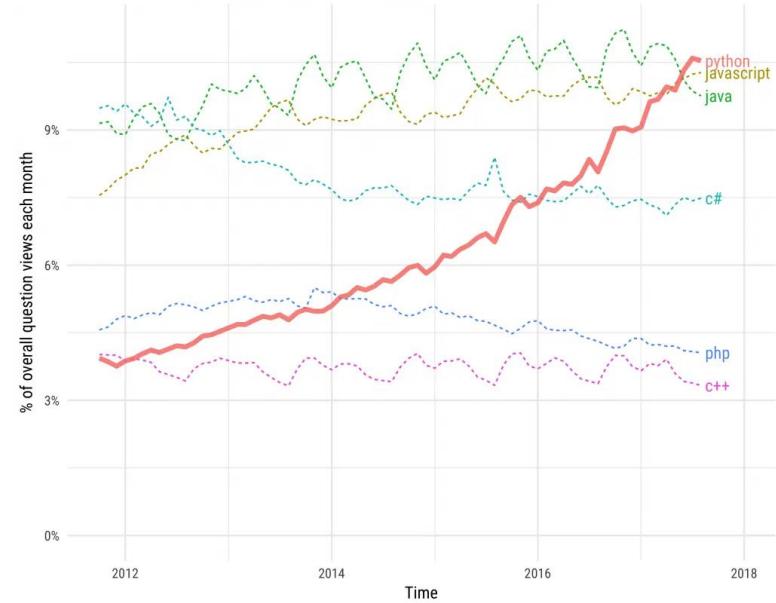
PYTHON



- A **high-level** programming language known for its simplicity and readability.
- Used in various domains such as web development, data analysis, **artificial intelligence**, and scientific computing.

Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries



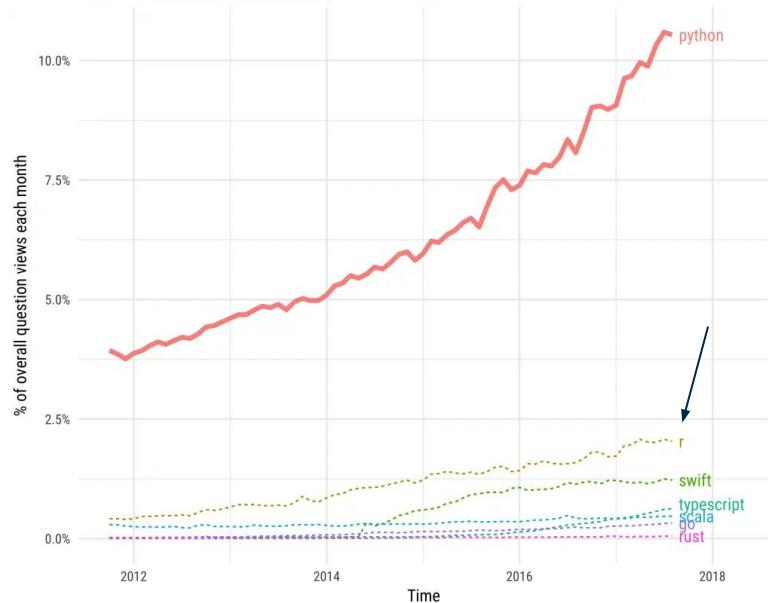
PYTHON



- A **high-level** programming language known for its simplicity and readability.
- Used in various domains such as web development, data analysis, **artificial intelligence**, and scientific computing.

Python compared to smaller, growing technologies

Based on question traffic in World Bank high-income countries



OUR SOFTWARE STACK



+

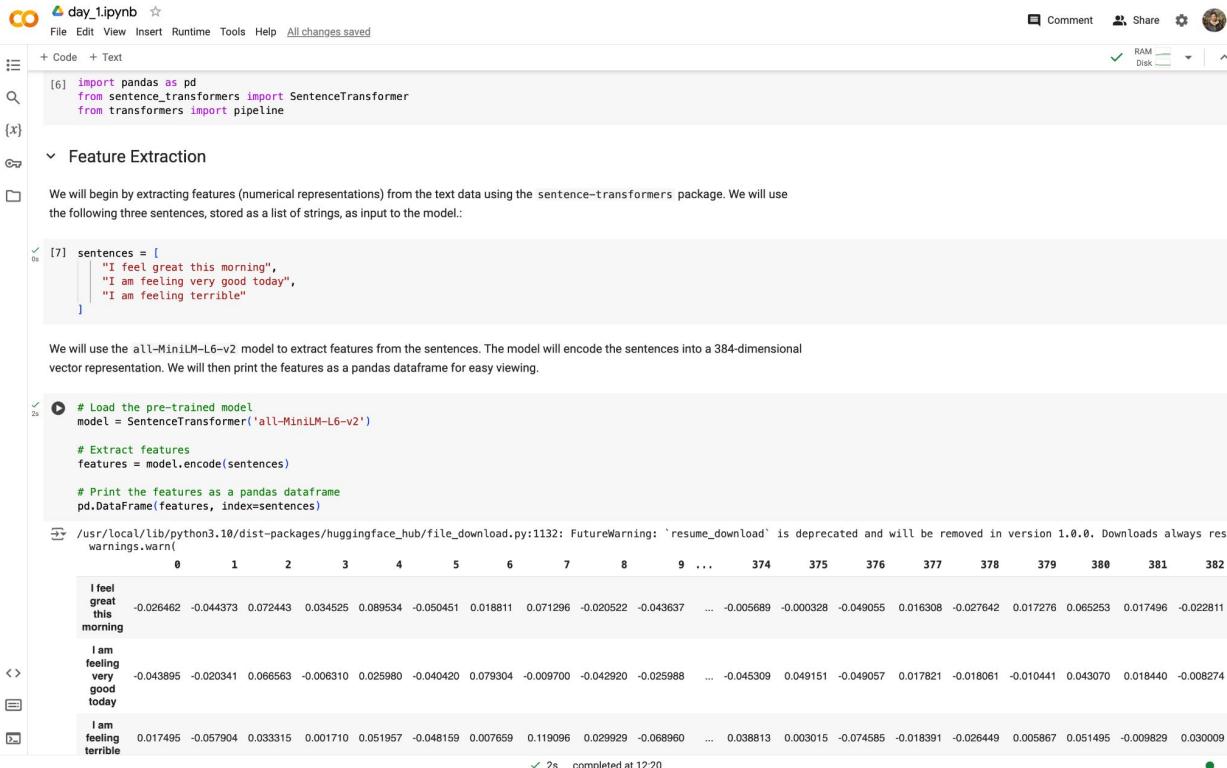


+



PYTHON  + Google Colab 

PYTHON + Google Colab



The image shows a screenshot of the Google Colab interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', 'Help', and a status message 'All changes saved'. To the right of the bar are 'Comment', 'Share', and settings icons. Below the bar, there are tabs for '+ Code' and '+ Text'. The main area contains code cells and text cells.

Code Cells:

```
[6] import pandas as pd
     from sentence_transformers import SentenceTransformer
     from transformers import pipeline
```

```
[7] sentences = [
      | "I feel great this morning",
      | "I am feeling very good today",
      | "I am feeling terrible"
    ]
```

Text Cell:

We will begin by extracting features (numerical representations) from the text data using the `sentence-transformers` package. We will use the following three sentences, stored as a list of strings, as input to the model:

Code Cells:

```
[25] # Load the pre-trained model
model = SentenceTransformer('all-MiniLM-L6-v2')

# Extract features
features = model.encode(sentences)

# Print the features as a pandas dataframe
pd.DataFrame(features, index=sentences)
```

Output:

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/file_download.py:1132: FutureWarning: 'resume_download' is deprecated and will be removed in version 1.0.0. Downloads always resume.
warnings.warn(
    0   1   2   3   4   5   6   7   8   9   ... 374 375 376 377 378 379 380 381 382
  I feel
  great
  this
  morning
-0.026462 -0.044373 0.072443 0.034525 0.089534 -0.050451 0.018811 0.071296 -0.020522 -0.043637 ... -0.005689 -0.000328 -0.049055 0.016308 -0.027642 0.017276 0.065253 0.017496 -0.022811

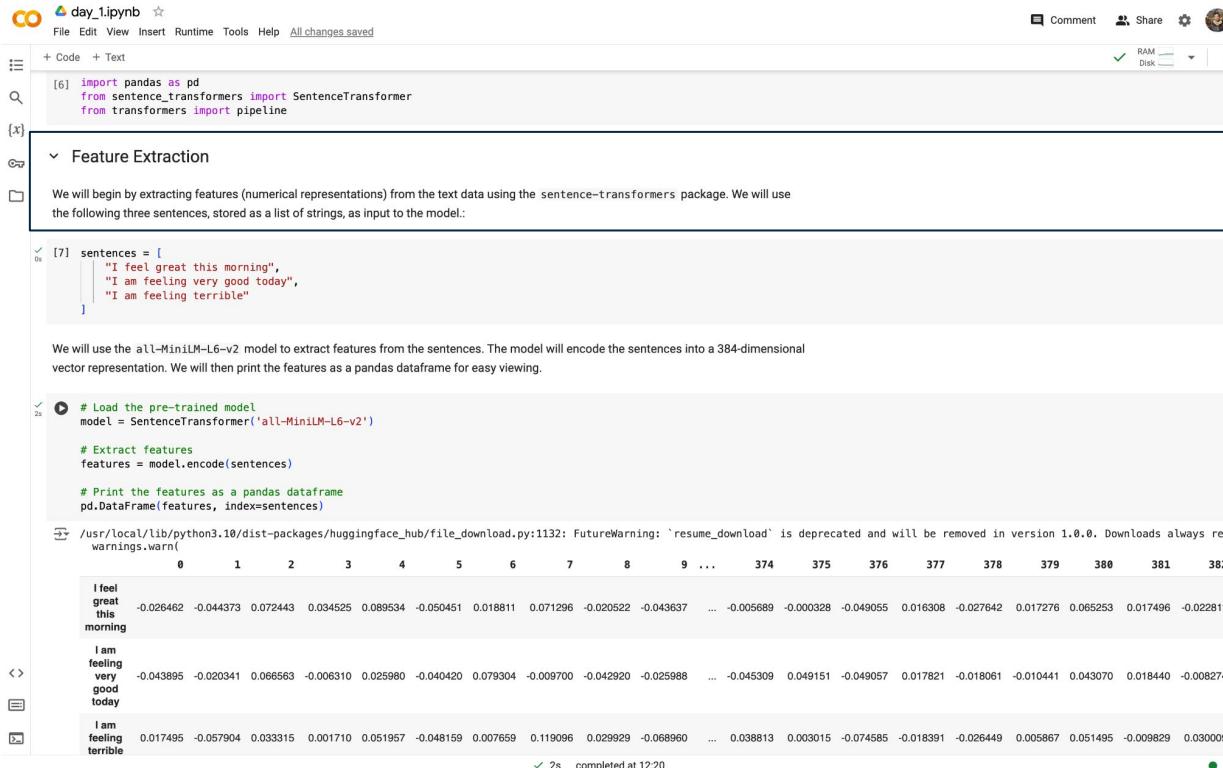
  I am
  feeling
  very
  good
  today
-0.043895 -0.020341 0.066563 -0.006310 0.025980 -0.040420 0.079304 -0.009700 -0.042920 -0.025988 ... -0.045309 0.049151 -0.049057 0.017821 -0.018061 -0.010441 0.043070 0.018440 -0.008274

  I am
  feeling
  terrible
0.017495 -0.057904 0.033315 0.001710 0.051957 -0.048159 0.007659 0.119096 0.029929 -0.068960 ... 0.038813 0.003015 -0.074585 -0.018391 -0.026449 0.005867 0.051495 -0.009829 0.030009
```

At the bottom, there's a progress bar indicating '2s completed at 12:20'.

PYTHON + Google Colab

1. Markdown



We will begin by extracting features (numerical representations) from the text data using the `sentence-transformers` package. We will use the following three sentences, stored as a list of strings, as input to the model:

```
[7] sentences = [
    "I feel great this morning",
    "I am feeling very good today",
    "I am feeling terrible"
]
```

We will use the `all-MiniLM-L6-v2` model to extract features from the sentences. The model will encode the sentences into a 384-dimensional vector representation. We will then print the features as a pandas dataframe for easy viewing.

```
[25] # Load the pre-trained model
model = SentenceTransformer('all-MiniLM-L6-v2')

# Extract features
features = model.encode(sentences)

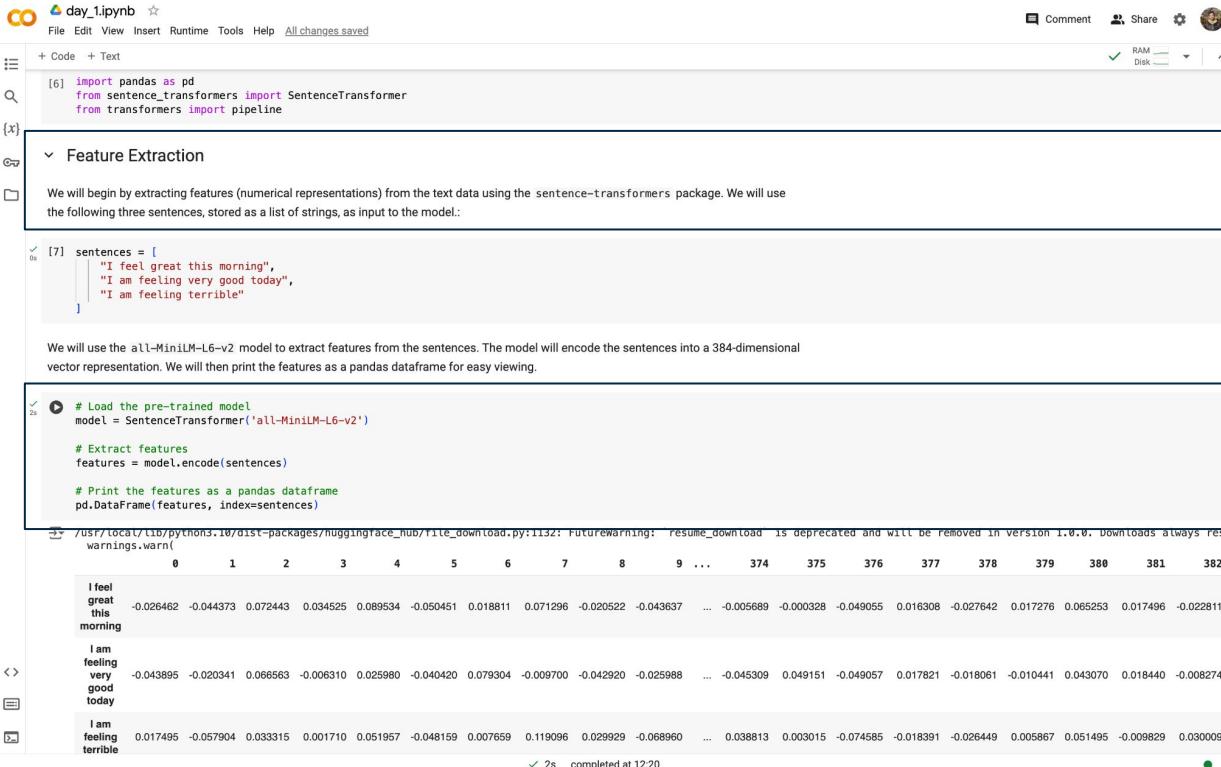
# Print the features as a pandas dataframe
pd.DataFrame(features, index=sentences)
```

/usr/local/lib/python3.10/dist-packages/huggingface_hub/file_download.py:1132: FutureWarning: 'resume_download' is deprecated and will be removed in version 1.0.0. Downloads always resume.

	0	1	2	3	4	5	6	7	8	9	...	374	375	376	377	378	379	380	381	382
I feel great this morning	-0.026462	-0.044373	0.072443	0.034525	0.089534	-0.050451	0.018811	0.071296	-0.020522	-0.043637	...	-0.005689	-0.000328	-0.049055	0.016308	-0.027642	0.017276	0.065253	0.017496	-0.022811
I am feeling very good today	-0.043895	-0.020341	0.066563	-0.006310	0.025980	-0.040420	0.079304	-0.009700	-0.042920	-0.025988	...	-0.045309	0.049151	-0.049057	0.017821	-0.018061	-0.010441	0.043070	0.018440	-0.008274
I am feeling terrible	0.017495	-0.057904	0.033315	0.001710	0.051957	-0.048159	0.007659	0.119096	0.029929	-0.068960	...	0.038813	0.003015	-0.074585	-0.018391	-0.026449	0.005867	0.051495	-0.009829	0.030009

✓ 2s completed at 12:20

PYTHON + Google Colab



The screenshot shows a Google Colab notebook titled "day_1.ipynb". The code cell at the top imports pandas and sentence-transformers, and defines a list of sentences. A section titled "Feature Extraction" explains that features will be extracted using the "all-MiniLM-L6-v2" model. The code cell below loads the model and encodes the sentences into a pandas DataFrame.

```
[6] import pandas as pd
     from sentence_transformers import SentenceTransformer
     from transformers import pipeline

{x}
  ✓ Feature Extraction

  We will begin by extracting features (numerical representations) from the text data using the sentence-transformers package. We will use the following three sentences, stored as a list of strings, as input to the model:

[7] sentences = [
    "I feel great this morning",
    "I am feeling very good today",
    "I am feeling terrible"
]

We will use the all-MiniLM-L6-v2 model to extract features from the sentences. The model will encode the sentences into a 384-dimensional vector representation. We will then print the features as a pandas dataframe for easy viewing.

[25] # Load the pre-trained model
model = SentenceTransformer('all-MiniLM-L6-v2')

# Extract features
features = model.encode(sentences)

# Print the features as a pandas dataframe
pd.DataFrame(features, index=sentences)
```

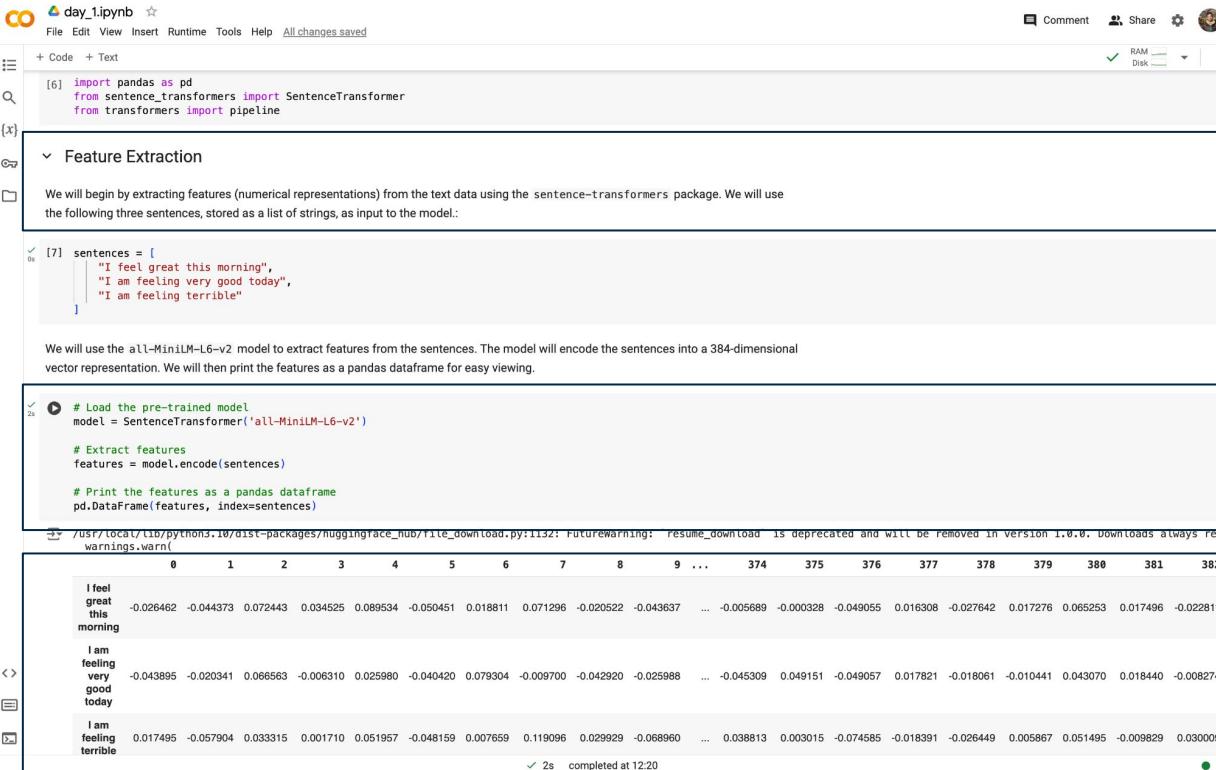
The output of the code cell shows a pandas DataFrame with 3 rows (sentences) and 382 columns (vector dimensions). The first row corresponds to "I feel great this morning", the second to "I am feeling very good today", and the third to "I am feeling terrible". The values are floating-point numbers ranging from -0.043895 to 0.017495.

	0	1	2	3	4	5	6	7	8	9	...	374	375	376	377	378	379	380	381	382
I feel great this morning	-0.026462	-0.044373	0.072443	0.034525	0.089534	-0.050451	0.018811	0.071296	-0.020522	-0.043637	...	-0.005689	-0.000328	-0.049055	0.016308	-0.027642	0.017276	0.065253	0.017496	-0.022811
I am feeling very good today	-0.043895	-0.020341	0.066563	-0.006310	0.025980	-0.040420	0.079304	-0.009700	-0.042920	-0.025988	...	-0.045309	0.049151	-0.049057	0.017821	-0.018061	-0.010441	0.043070	0.018440	-0.008274
I am feeling terrible	0.017495	-0.057904	0.033315	0.001710	0.051957	-0.048159	0.007659	0.119096	0.029929	-0.068960	...	0.038813	0.003015	-0.074585	-0.018391	-0.026449	0.005867	0.051495	-0.009829	0.030009

1. Markdown

2. Code

PYTHON + Google Colab



The screenshot shows a Google Colab notebook titled "day_1.ipynb". The interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help) and a toolbar with Comment, Share, and settings. A sidebar on the left shows sections like Feature Extraction, which is currently expanded. The main area contains code cells and their outputs.

1. Markdown

We will begin by extracting features (numerical representations) from the text data using the `sentence-transformers` package. We will use the following three sentences, stored as a list of strings, as input to the model:

```
[7] sentences = [
    "I feel great this morning",
    "I am feeling very good today",
    "I am feeling terrible"
]
```

We will use the `all-MiniLM-L6-v2` model to extract features from the sentences. The model will encode the sentences into a 384-dimensional vector representation. We will then print the features as a pandas dataframe for easy viewing.

```
[25] # Load the pre-trained model
model = SentenceTransformer('all-MiniLM-L6-v2')

# Extract features
features = model.encode(sentences)

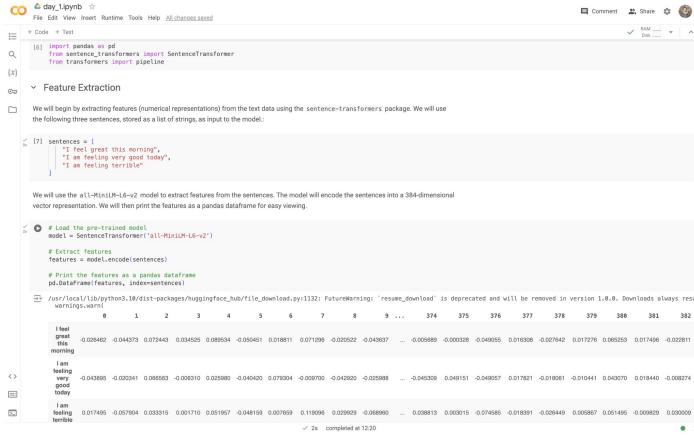
# Print the features as a pandas dataframe
pd.DataFrame(features, index=sentences)
```

2. Code

3. Printouts

	0	1	2	3	4	5	6	7	8	9	...	374	375	376	377	378	379	380	381	382
I feel great this morning	-0.026462	-0.044373	0.072443	0.034525	0.089534	-0.050451	0.018811	0.071296	-0.020522	-0.043637	...	-0.005689	-0.000328	-0.049055	0.016308	-0.027642	0.017276	0.065253	0.017496	-0.022811
I am feeling very good today	-0.043895	-0.020341	0.066563	-0.006310	0.025980	-0.040420	0.079304	-0.009700	-0.042920	-0.025988	...	-0.045309	0.049151	-0.049057	0.017821	-0.018061	-0.010441	0.043070	0.018440	-0.008274
I am feeling terrible	0.017495	-0.057904	0.033315	0.001710	0.051957	-0.048159	0.007659	0.119096	0.029929	-0.068960	...	0.038813	0.003015	-0.074585	-0.018391	-0.026449	0.005867	0.051495	-0.009829	0.030009

PYTHON + Google Colab



A screenshot of a Google Colab notebook titled "6-day_tutorial.ipynb". The code cell contains Python code for sentiment analysis using the SentenceTransformers library. It imports pandas and SentenceTransformer, and defines a pipeline. It then extracts features from three sentences: "I am great this morning", "I am feeling very good today", and "I am feeling terrible". The output shows the extracted 384-dimensional vector features as a pandas DataFrame.

```
# Feature Extraction
# We will begin by extracting features (numerical representations) from the text data using the sentence-transformers package. We will use the following three sentences, stored as a list of strings, as input to the model.

sentences = ["I am great this morning",
             "I am feeling very good today",
             "I am feeling terrible"]

We will use the 'all-MiniLM-L6-v2' model to extract features from the sentences. The model will encode the sentences into a 384-dimensional vector representation. We will then print the features as a pandas DataFrame for easy viewing.

# Load the pre-trained model.
model = SentenceTransformer('all-MiniLM-L6-v2')

# Extract features
features = model.encode(sentences)

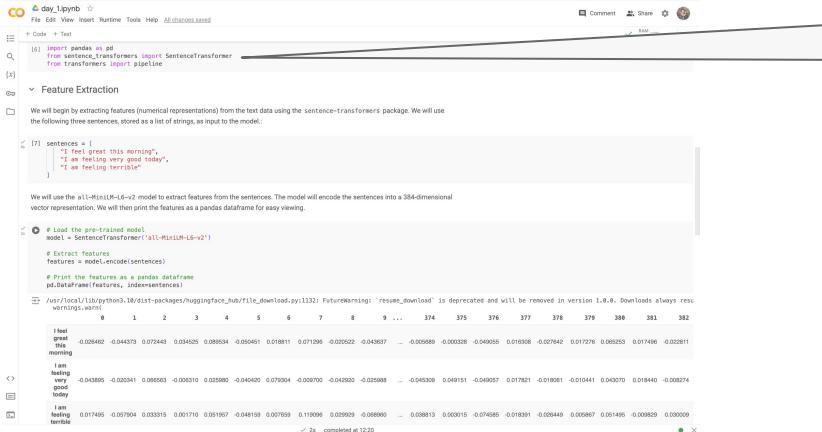
# Print the features as a pandas DataFrame
pd.DataFrame(features, index=sentences)
```

/usr/local/lib/python3.10/dist-packages/huggingface_hub/file_download.py:132: FutureWarning: 'resume_download' is deprecated and will be removed in version 1.8.0. Downloads always resume.

	0	1	2	3	4	5	6	7	8	9	374	375	376	377	378	379	380	381	382
I am great this morning	-0.036462	-0.044370	0.072443	0.034625	0.036954	-0.054610	0.019811	0.012096	-0.020262	-0.044807	... -0.005889	-0.003328	-0.040055	0.016326	-0.027462	0.017276	0.063253	0.074696	-0.028811
I am feeling very good today	-0.043865	-0.032041	0.085583	-0.005310	0.023590	-0.040400	0.079304	-0.009700	-0.042920	-0.025088	... -0.040309	0.049151	-0.049057	0.017821	-0.010981	-0.010441	0.043070	0.018440	-0.008274
I am feeling terrible	0.017495	-0.057904	0.033515	0.001710	0.051957	-0.048159	0.007669	0.119096	0.020909	-0.086990	... 0.038813	0.000015	-0.074586	-0.016891	0.009449	0.008967	0.051465	-0.009829	0.030009

PYTHON + Google Colab

package imports



```
# Code + Test
File Edit View Insert Runtime Help All changes saved
[1]: import pandas as pd
from sentence_transformers import SentenceTransformer
from transformers import pipeline
(x)
Feature Extraction
We will begin by extracting features (numerical representations) from the text data using the sentence-transformers package. We will use the following three sentences, stored as a list of strings, as input to the model.
[1]: sentences = [
    "I am great this morning",
    "I am feeling very good today",
    "I am feeling terrible"
]
We will use the all-MiniLM-L6-v2 model to extract features from the sentences. The model will encode the sentences into a 384-dimensional vector representation. We will then print the features as a pandas dataframe for easy viewing.
[2]: # Load the pre-trained model
model = SentenceTransformer('all-MiniLM-L6-v2')

# Extract features
features = model.encode(sentences)

# Print the features as a pandas dataframe
pd.DataFrame(features, index=sentences)
/notebook/python3.10/dist-packages/huggingface_hub/file_download.py:132: FutureWarning: 'resume_download' is deprecated and will be removed in version 1.8.0. Downloads always resume.
warnings.warn(
    0   1   2   3   4   5   6   7   8   9 ... 374 375 376 377 378 379 380 381 382
  I am
great
this
morning
  I am
feeling
very
good
today
  I am
feeling
terrible
  0.017495 -0.057904 0.033515 0.001710 0.051957 -0.048159 0.007669 0.119098 ... 0.039813 0.000015 -0.074586 -0.018891 0.009449 0.009867 0.051465 -0.009829 0.030000
  ✓ 2s completed at 12:20
```

```
import pandas as pd
from sentence_transformers import SentenceTransformer
from transformers import pipeline
```

PYTHON + Google Colab

A screenshot of a Google Colab notebook titled "6-day-lipynb". The code cell contains imports for pandas, SentenceTransformer, and pipeline, followed by a list of sentences and a feature extraction process using the all-MiniLM-L6-v2 model.

```
import pandas as pd
from sentence_transformers import SentenceTransformer
from transformers import pipeline

sentences = [
    "I feel great this morning",
    "I am feeling very good today",
    "I am feeling terrible"
]

We will begin by extracting features (numerical representations) from the text data using the sentence-transformers package. We will use the following three sentences, stored as a list of strings, as input to the model.

# Load the pre-trained model.
model = SentenceTransformer('all-MiniLM-L6-v2')

# Extract features
features = model.encode(sentences)

# Print the features as a pandas dataframe
pd.DataFrame(features, index=sentences)
```

The output shows the extracted 384-dimensional vector representation for each sentence, with columns ranging from 0 to 382.

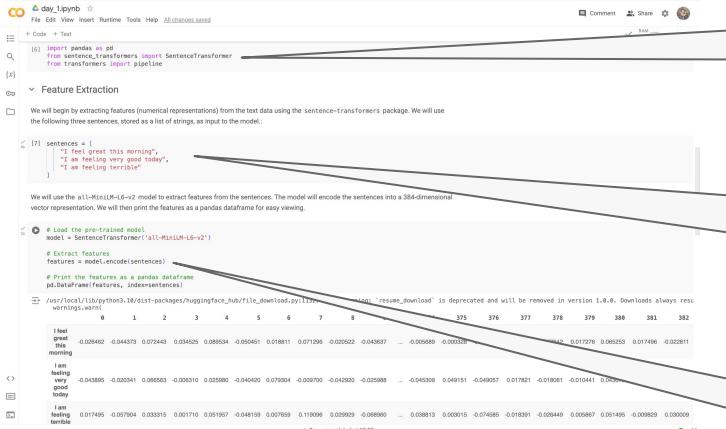
package imports

```
import pandas as pd
from sentence_transformers import SentenceTransformer
from transformers import pipeline
```

variable assignment, lists, strings

```
sentences = [
    "I feel great this morning",
    "I am feeling very good today",
    "I am feeling terrible"
]
```

PYTHON + Google Colab



A screenshot of a Google Colab notebook titled "6-day-lipynn". The code cell contains imports for pandas, SentenceTransformer, and pipeline. The output cell shows three sentences: "I feel great this morning", "I am feeling very good today", and "I am feeling terrible". Below the output, a warning about the 'resume_download' parameter is shown. The code then loads a pre-trained model ('all-MiniLM-L6-v2'), encodes the sentences, and prints the resulting features as a pandas DataFrame.

```
import pandas as pd
from sentence_transformers import SentenceTransformer
from transformers import pipeline

sentences = [
    "I feel great this morning",
    "I am feeling very good today",
    "I am feeling terrible"
]

# Load the pre-trained model
model = SentenceTransformer('all-MiniLM-L6-v2')

# Extract features
features = model.encode(sentences)

# Print the features as a pandas DataFrame
pd.DataFrame(features, index=sentences)
```

Warning: /usr/local/lib/python3.10/dist-packages/huggingface_hub/file_download.py:123: UserWarning: 'resume_download' is deprecated and will be removed in version 1.8.0. Downloads always resume now!

	0	1	2	3	4	5	6	7	8	375	376	377	378	379	380	381	382		
I feel great this morning	-0.036462	-0.043701	0.072483	0.034625	0.036954	-0.054651	0.019811	0.071298	-0.026262	-0.048087	-0.005889	-0.003231	-0.046309	0.049151	-0.049057	0.017821	-0.019081	-0.010441	0.0050
I am feeling very good today	-0.043895	-0.020341	0.085583	-0.005310	0.025980	-0.040402	0.079304	-0.009700	-0.042920	-0.023588	... -0.046309	0.049151	-0.049057	0.017821	-0.019081	-0.010441	0.0050		
I am feeling terrible	0.017495	-0.057904	0.033515	0.001710	0.051957	-0.048159	0.097669	0.119098	0.029609	-0.086990	... 0.058813	0.000015	0.074586	-0.018891	0.005449	0.008967	0.051465	-0.009829	0.000000

package imports

```
import pandas as pd
from sentence_transformers import SentenceTransformer
from transformers import pipeline
```

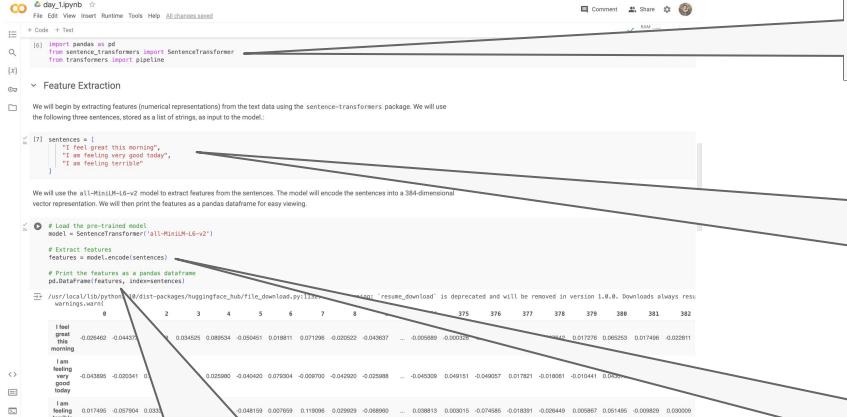
variable assignment, lists, strings

```
sentences = [
    "I feel great this morning",
    "I am feeling very good today",
    "I am feeling terrible"
]
```

dot notation, methods, attributes

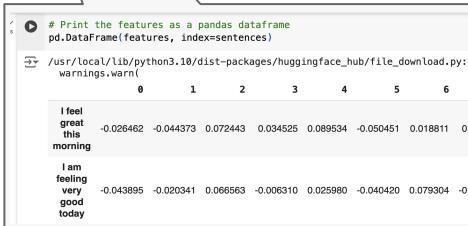
```
# Extract features
features = model.encode(sentences)
```

PYTHON + Google Colab



A screenshot of a Google Colab notebook titled '6-day_tutorial.ipynb'. The code cell contains imports for pandas, sentence-transformers, and transformers, along with a pipeline definition. The output cell shows three sentences: "I feel great this morning", "I am feeling very good today", and "I am feeling terrible". Below the output, a warning about the 'resume_download' method is visible.

printing



A screenshot of a Google Colab notebook showing the output of a print statement. It displays a pandas DataFrame with columns labeled 0 through 6, containing numerical values for each of the three sentences.

```
import pandas as pd
from sentence_transformers import SentenceTransformer
from transformers import pipeline
```

package imports

variable assignment, lists, strings

```
sentences = [
    "I feel great this morning",
    "I am feeling very good today",
    "I am feeling terrible"
]
```

dot notation, methods, attributes

```
# Extract features
features = model.encode(sentences)
```

OUR SOFTWARE STACK



+



+



Time to install
our stack...



MAX PLANCK INSTITUTE
FOR HUMAN DEVELOPMENT





Code



Issues



Pull requests



Actions



Projects



Wiki



Security



Insights



Settings



LLM4BeSci_Ljubljana2026

Public



Pin



0



0



Star



main

1 Branch

0 Tags

Go to file



Add file

Code

About



Zak-Hussain slide upload note

a18ea0b · 9 hours ago 26 Commits



day_1

use latest day_1.ipynb

last week



day_2

typo

10 hours ago



day_3

typo

9 hours ago



day_4

typo

9 hours ago



day_5

update with newer justaism paper version

9 hours ago



.gitignore

Initial commit

last week



LICENSE.txt

add LICENSE.txt

last week



README.md

slide upload note

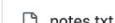
9 hours ago



cover_gserm.png

Add files via upload

last week



notes.txt

add notes.txt

last week

README License



LLM4BeSci at GSERM, Ljubljana 2026

Applying open-

The course introduces the use of open large language models (LLMs) from the Hugging Face ecosystem for research in the behavioral and social sciences.

Readme

View license

Activity

0 stars

0 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Contributors 2

Zak-Hussain Zak Hussain

dwulff Dirk Wulff

Languages