Intro Hugging Face and Python

Dirk Wulff & Zak Hussain

MAX PLANCK INSTITUTE
FOR HUMAN DEVELOPMENT

UNI
BASEL

OUR SOFTWARE STACK

python™ + + Google colab

OUR SOFTWARE STACK

python™ + + Google colab

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

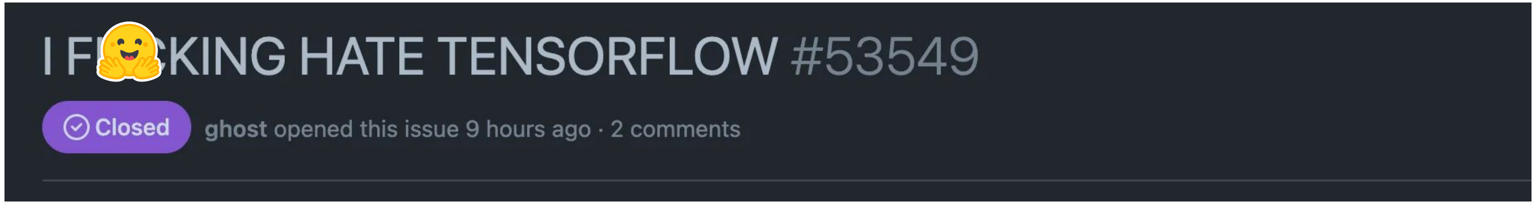
WHY IS HUGGING FACE?

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


I F🤗CKING HATE TENSORFLOW #53549

ghost opened this issue 9 hours ago · 2 comments

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')
```

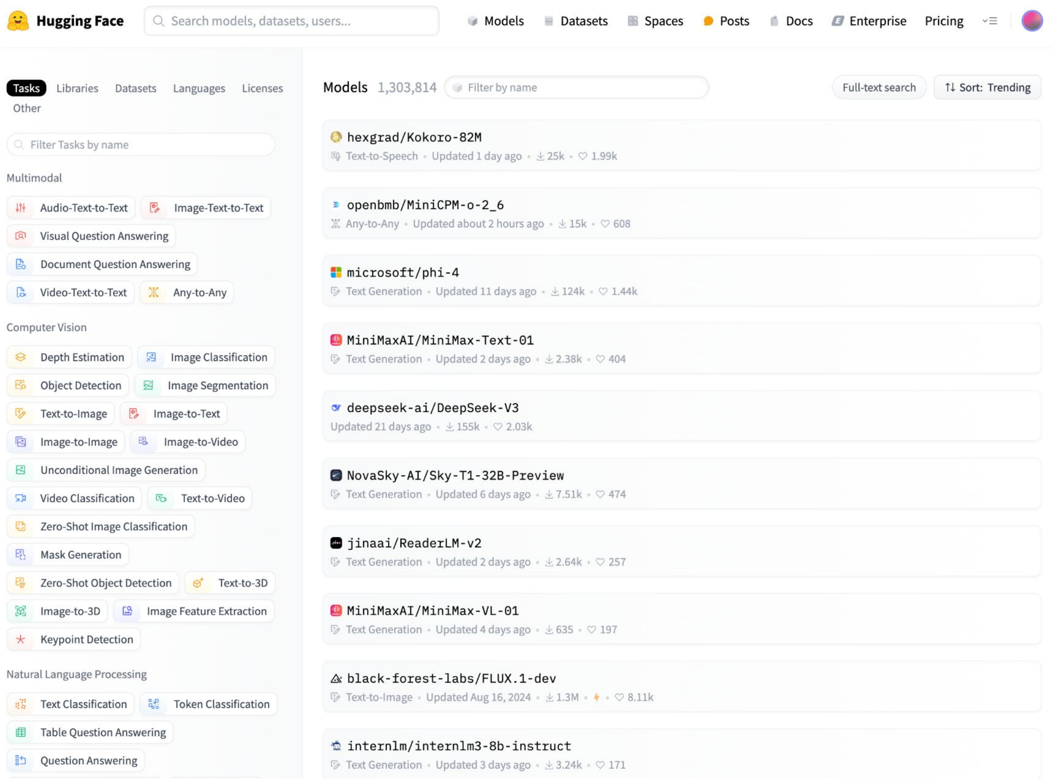
Load model input

```
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



Hugging Face | Search models, datasets, users... | Models | Datasets | Spaces | Posts | Docs | Enterprise | Pricing

Tasks | Libraries | Datasets | Languages | Licenses | Other

Filter Tasks by name

Multimodal

- Audio-Text-to-Text
- Image-Text-to-Text
- Visual Question Answering
- Document Question Answering
- Video-Text-to-Text
- Any-to-Any

Computer Vision

- Depth Estimation
- Image Classification
- Object Detection
- Image Segmentation
- Text-to-Image
- Image-to-Text
- Image-to-Image
- Image-to-Video
- Unconditional Image Generation
- Video Classification
- Text-to-Video
- Zero-Shot Image Classification
- Mask Generation
- Zero-Shot Object Detection
- Text-to-3D
- Image-to-3D
- Image Feature Extraction
- Keypoint Detection

Natural Language Processing

- Text Classification
- Token Classification
- Table Question Answering
- Question Answering

Models 1,303,814 | Filter by name | Full-text search | Sort: Trending

hexgrad/Kokoro-82M
Text-to-Speech • Updated 1 day ago • 25k • 1.99k

openbmb/MiniCPM-o-2_6
Any-to-Any • Updated about 2 hours ago • 15k • 608

microsoft/phi-4
Text Generation • Updated 11 days ago • 124k • 1.44k

MiniMaxAI/MiniMax-Text-01
Text Generation • Updated 2 days ago • 2.38k • 404

deepseek-ai/DeepSeek-V3
Updated 21 days ago • 155k • 2.03k

NovaSky-AI/Sky-T1-32B-Preview
Text Generation • Updated 6 days ago • 7.51k • 474

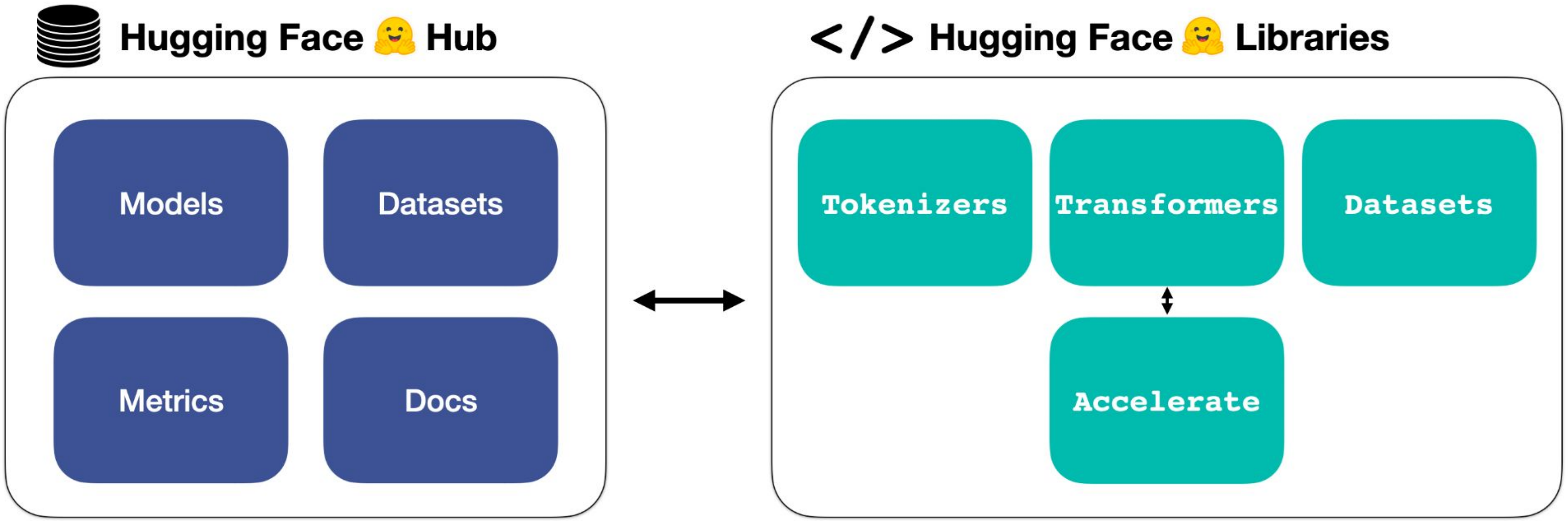
jinaai/ReaderLM-v2
Text Generation • Updated 2 days ago • 2.64k • 257

MiniMaxAI/MiniMax-VL-01
Text Generation • Updated 4 days ago • 635 • 197

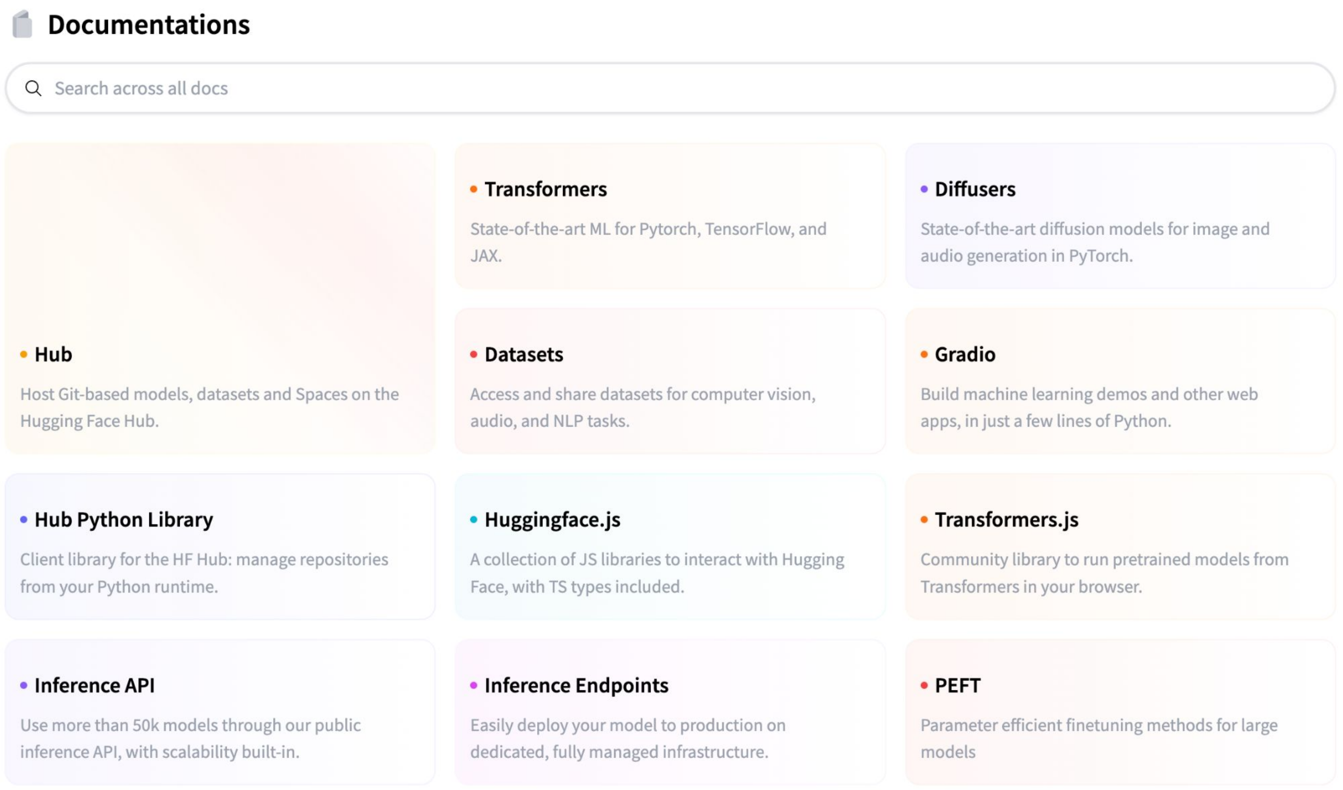
black-forest-labs/FLUX.1-dev
Text-to-Image • Updated Aug 16, 2024 • 1.3M • 8.11k

internlm/internlm3-8b-instruct
Text Generation • Updated 3 days ago • 3.24k • 171

THE HUGGING FACE ECOSYSTEM



HUGGING FACE DOCUMENTATION



Documentations

Search across all docs

- **Hub**
  Host Git-based models, datasets and Spaces on the Hugging Face Hub.
- **Transformers**
  State-of-the-art ML for Pytorch, TensorFlow, and JAX.
- **Diffusers**
  State-of-the-art diffusion models for image and audio generation in PyTorch.
- **Datasets**
  Access and share datasets for computer vision, audio, and NLP tasks.
- **Gradio**
  Build machine learning demos and other web apps, in just a few lines of Python.
- **Hub Python Library**
  Client library for the HF Hub: manage repositories from your Python runtime.
- **Huggingface.js**
  A collection of JS libraries to interact with Hugging Face, with TS types included.
- **Transformers.js**
  Community library to run pretrained models from Transformers in your browser.
- **Inference API**
  Use more than 50k models through our public inference API, with scalability built-in.
- **Inference Endpoints**
  Easily deploy your model to production on dedicated, fully managed infrastructure.
- **PEFT**
  Parameter efficient finetuning methods for large models

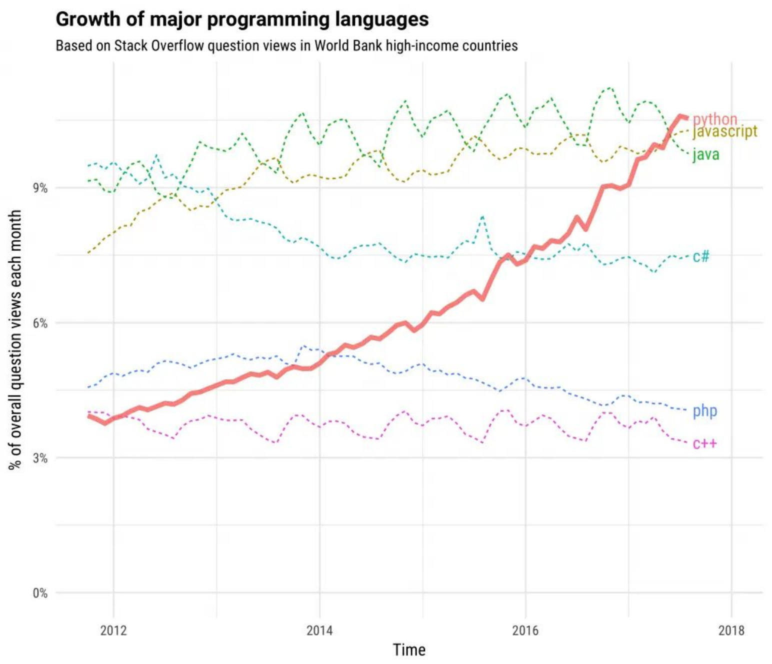
OUR SOFTWARE STACK

python™ + + Google colab

PYTHON

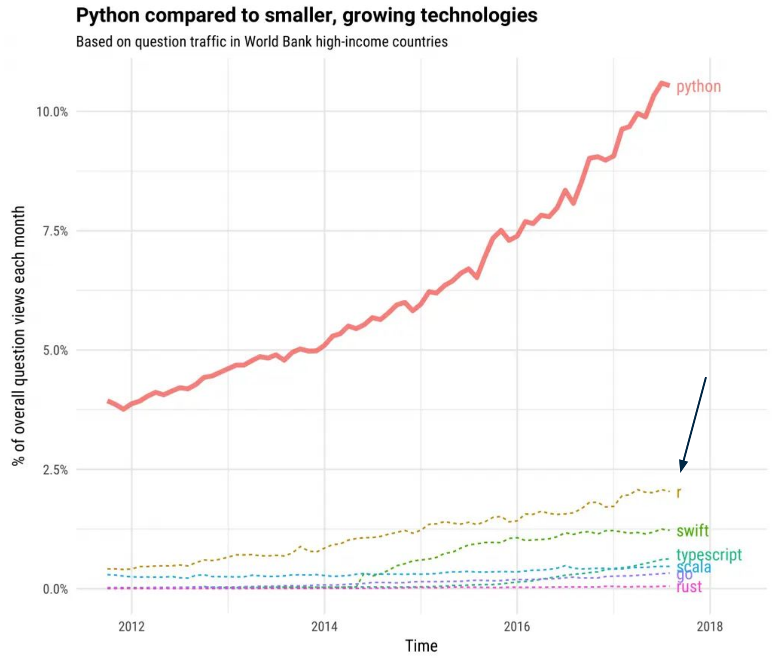
PYTHON

- 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

- 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.

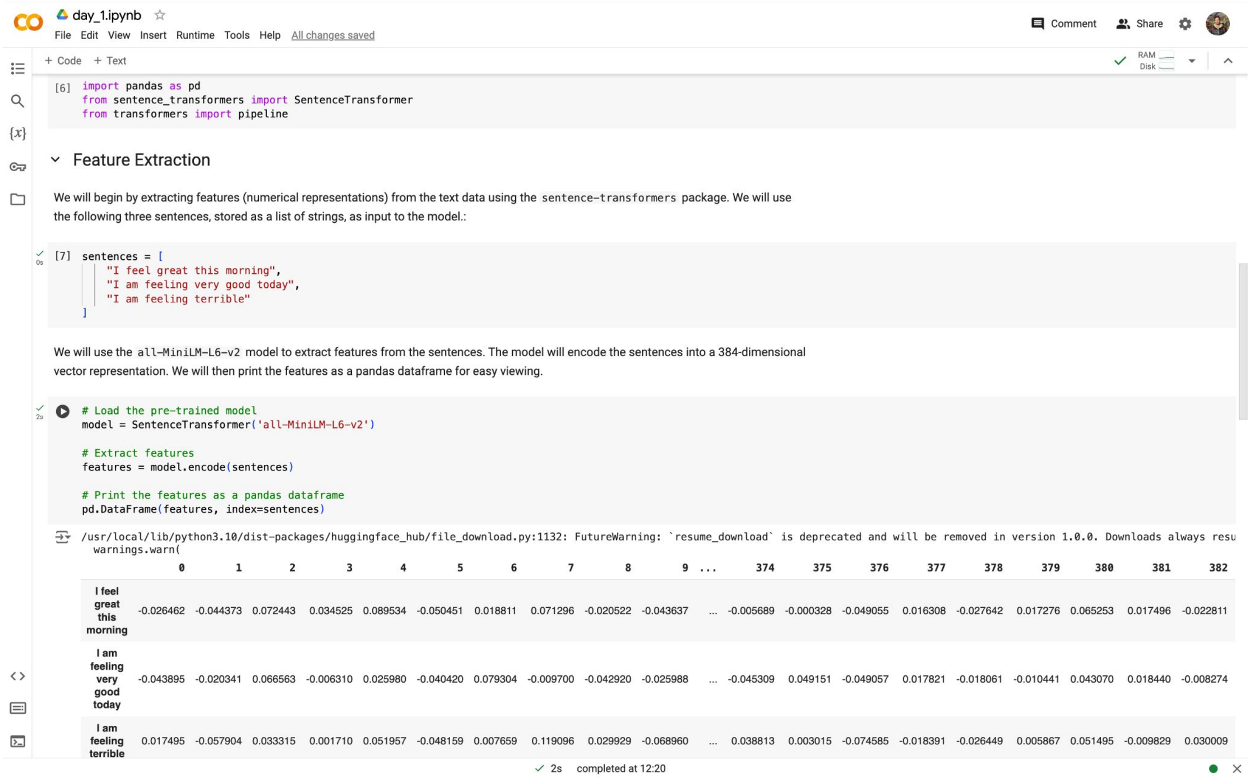


OUR SOFTWARE STACK

python™ + + Google colab

PYTHON + Google Colab

PYTHON + Google Colab



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

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.

```
# 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)
```

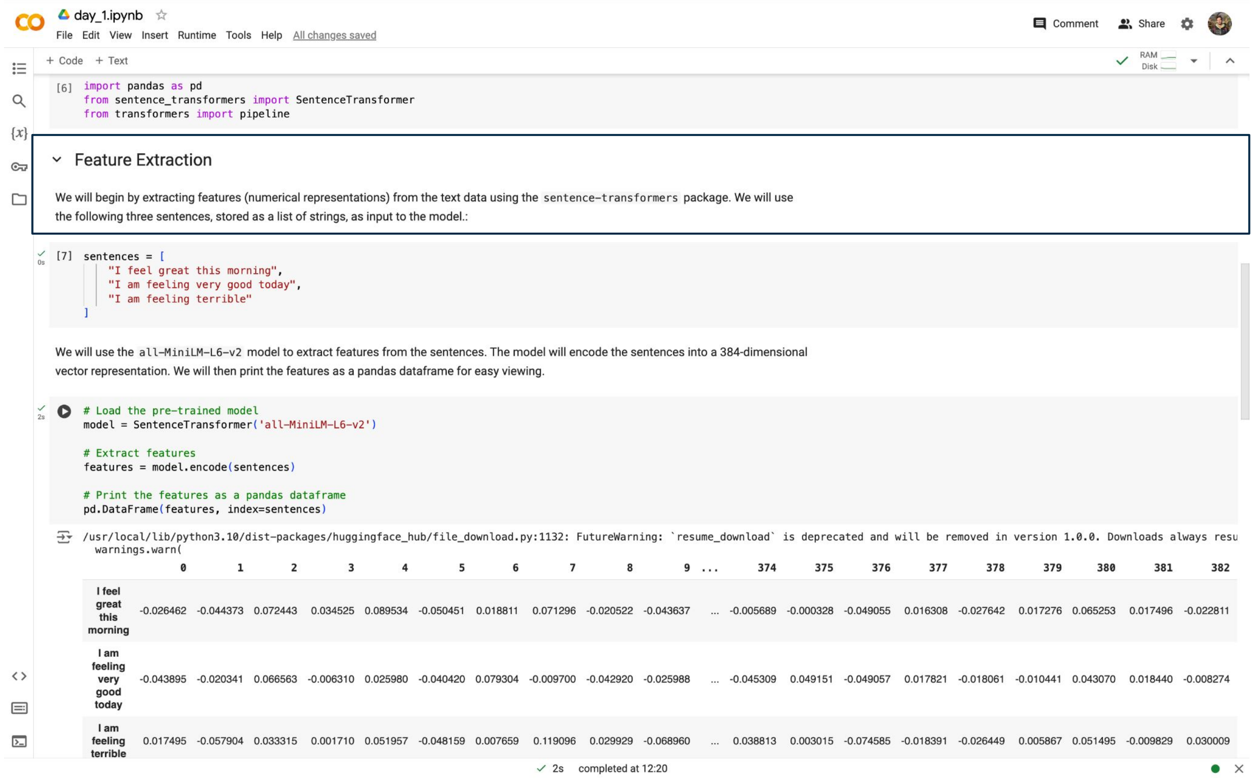
```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/file_download.py:1132: FutureWarning: `resume_download` is deprecated and will be removed in version 1.0.0. Downloads always resu
  warnings.warn(
```

	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

✓ 2s completed at 12:20

PYTHON + Google Colab

1. Markdown



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

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.

```
# 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 resu
  warnings.warn(
```

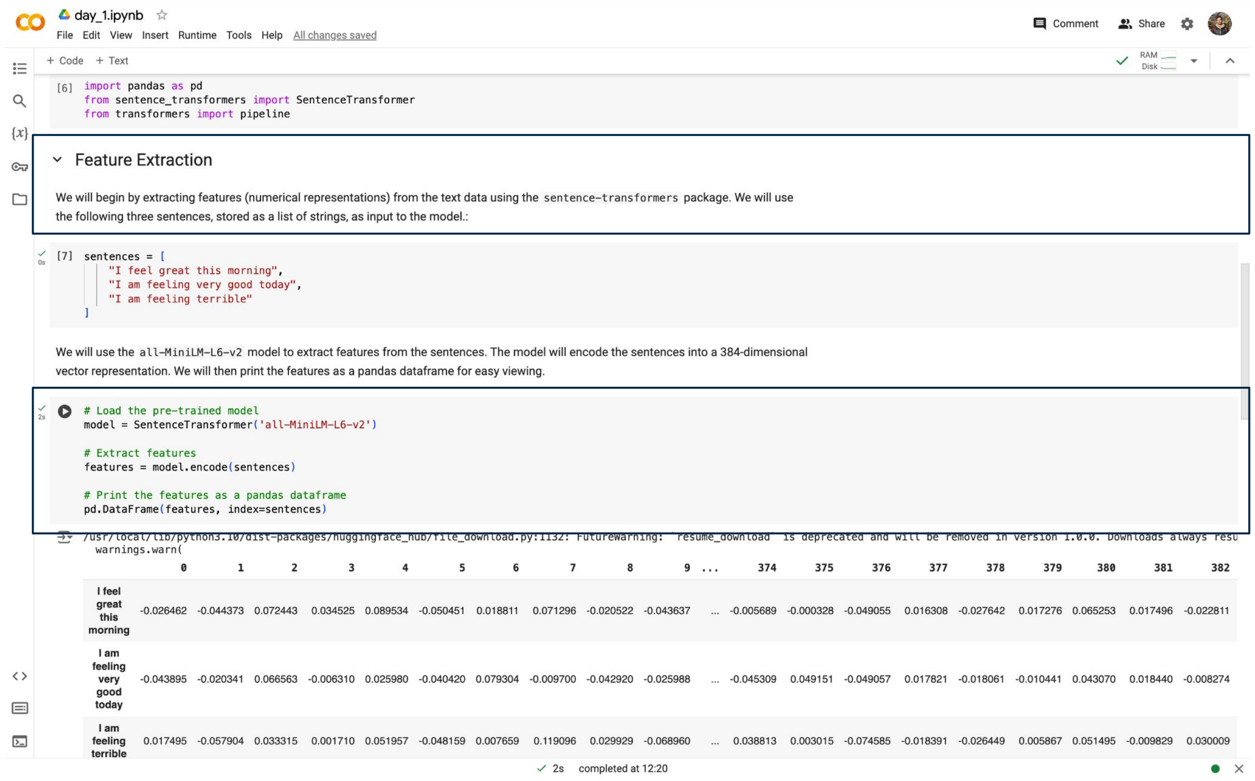
	0	1	2	3	4	5	6	7	8	9	...	374	375	376	377	378	379	380	381	382
I feel great this morning	-0.026462	-0.044373	0.072443	0.034525	0.089534	-0.050451	0.018811	0.071296	-0.020522	-0.043637	...	-0.005689	-0.000328	-0.049055	0.016308	-0.027642	0.017276	0.065253	0.017496	-0.022811
I am feeling very good today	-0.043895	-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

✓ 2s completed at 12:20

PYTHON + Google Colab

1. Markdown

2. Code



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

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.

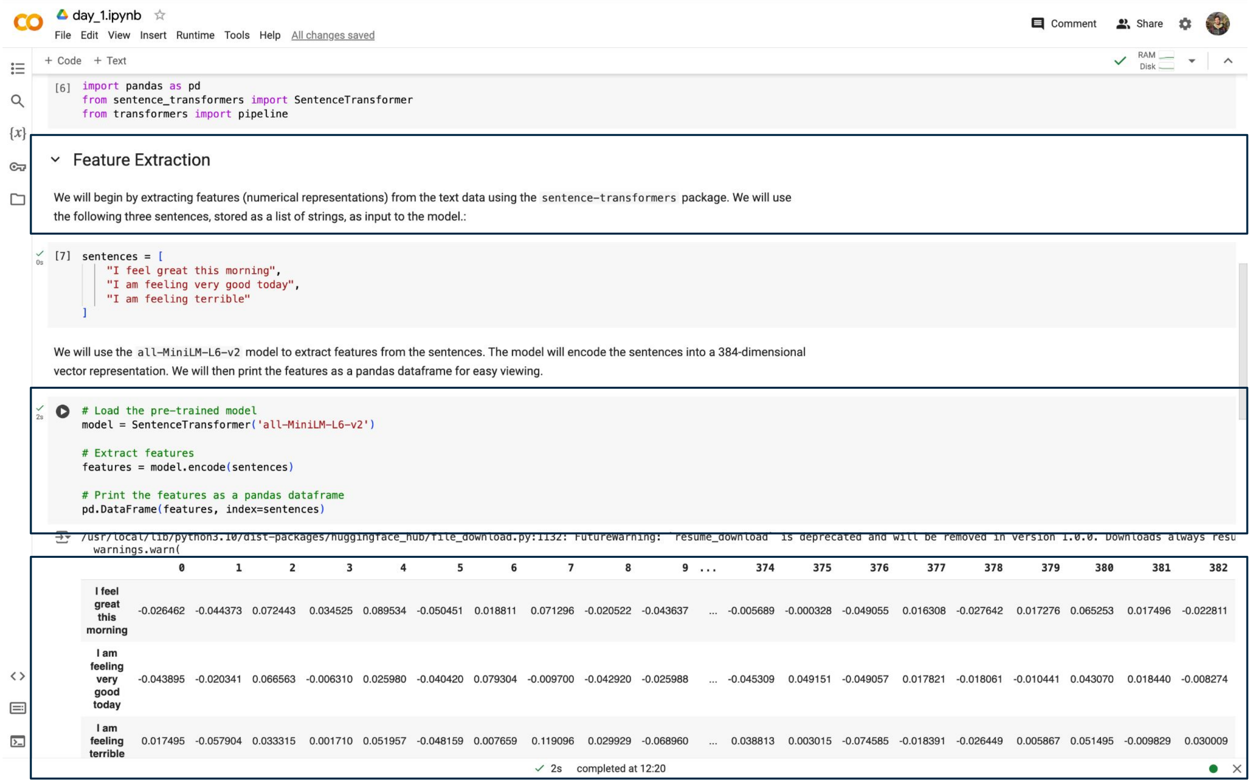
```
# Load the pre-trained model
model = SentenceTransformer('all-MiniLM-L6-v2')

# 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 resu
  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

✓ 2s completed at 12:20

PYTHON + Google Colab



1. Markdown

2. Code

3. Printouts

day_1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

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

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.

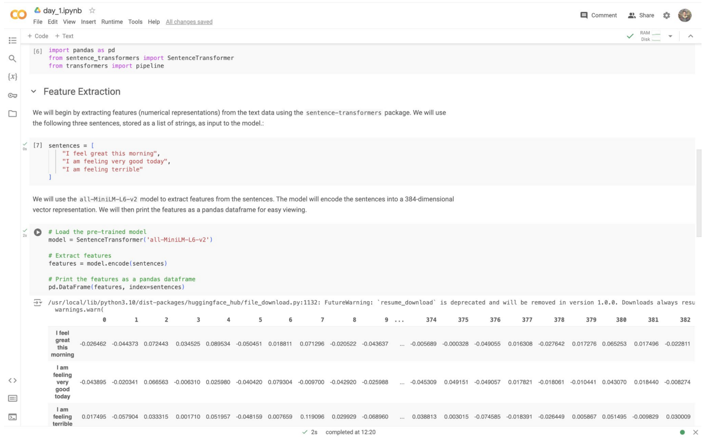
```
# Load the pre-trained model
model = SentenceTransformer('all-MiniLM-L6-v2')

# 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 resu
  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

✓ 2s completed at 12:20

PYTHON + Google Colab



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

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.

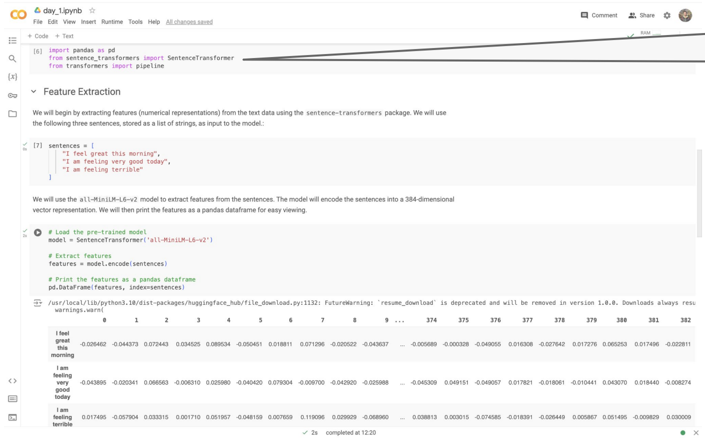
```
# Load the pre-trained model
model = SentenceTransformer('all-MiniLM-L6-v2')

# 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 resu
  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

✓ 2s completed at 12:20

PYTHON + Google Colab



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

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.

```
# 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 resu
  warnings.warn(
```

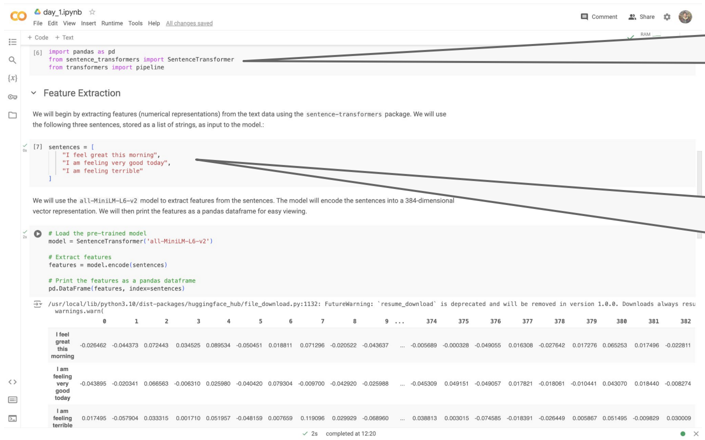
	0	1	2	3	4	5	6	7	8	9	...	374	375	376	377	378	379	380	381	382
I feel great this morning	-0.026462	-0.044373	0.072443	0.034525	0.089534	-0.050451	0.018811	0.071296	-0.020522	-0.043637	...	-0.005689	-0.000328	-0.049055	0.016308	-0.027642	0.017276	0.065253	0.017496	-0.022811
I am feeling very good today	-0.043895	-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

✓ 2s completed at 12:20

package imports

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

PYTHON + Google Colab



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

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.

```
# 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 resu
  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

✓ 2s completed at 12:20

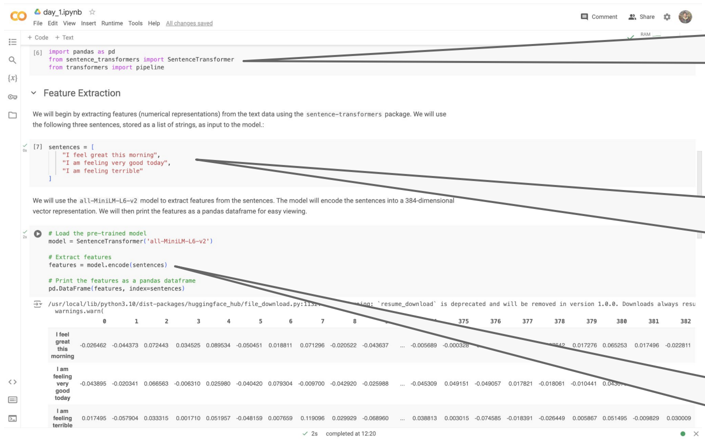
package imports

```
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



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

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.

```
# 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:113[illegible] [illegible]ning: `resume_download` is deprecated and will be removed in version 1.0.0. Downloads always resu
  warnings.warn(
```

	0	1	2	3	4	5	6	7	8	[illegible]	...	[illegible]	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	[illegible]	[illegible]	[illegible]	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	[illegible]	[illegible]	[illegible]
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

package imports

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

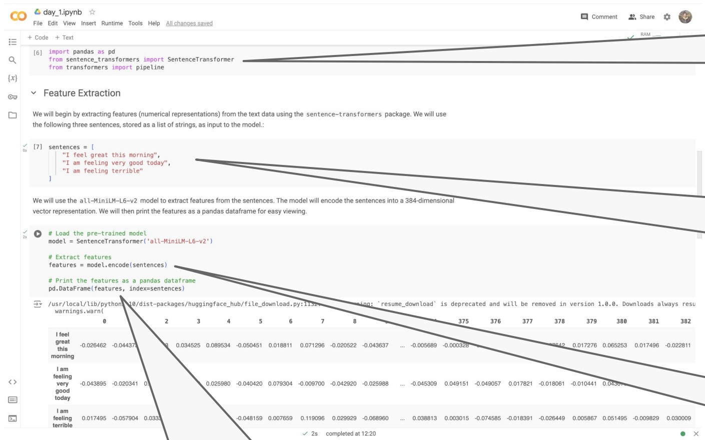
variable assignment, lists, strings

```
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



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

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.:

```python
[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.

```python
# 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/python[illegible]10/dist-packages/huggingface_hub/file_download.py:113[illegible] [illegible]ning: `resume_download` is deprecated and will be removed in version 1.0.0. Downloads always resu
  warnings.warn(
                    0          [illegible]  2  3         4          5          6          7          8          [illegible]  ...  [illegible]  375        376        377        378        379        380        381        382
I feel great     -0.026462  -0.04437[illegible]  [illegible]  0.034525   0.089534  -0.050451   0.018811   0.071296  -0.020522  -0.043637  ...  -0.005689  -0.000328  [illegible]  [illegible]  [illegible]7642  0.017276  0.065253  0.017496  -0.022811
this morning
I am feeling     -0.043895  -0.020341  0[illegible]  [illegible]  0.025980  -0.040420   0.079304  -0.009700  -0.042920  -0.025988  ...  -0.045309   0.049151  -0.049057   0.017821  -0.018061  -0.010441   0.0430[illegible]
very good today
I am feeling      0.017495  -0.057904   0.033[illegible]  [illegible]  [illegible]  -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
terrible
```

✓ 2s completed at 12:20

package imports

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

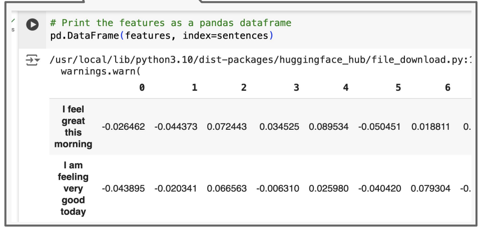
variable assignment, lists, strings

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

dot notation, methods, attributes

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

printing



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

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/file_download.py:
  warnings.warn(
                          0          1          2          3          4          5          6
I feel great this   -0.026462  -0.044373   0.072443   0.034525   0.089534  -0.050451   0.018811  0.
morning
I am feeling very   -0.043895  -0.020341   0.066563  -0.006310   0.025980  -0.040420   0.079304  -0.
good today
```

Dirk Wulff & Zak Hussain UniBas 2025

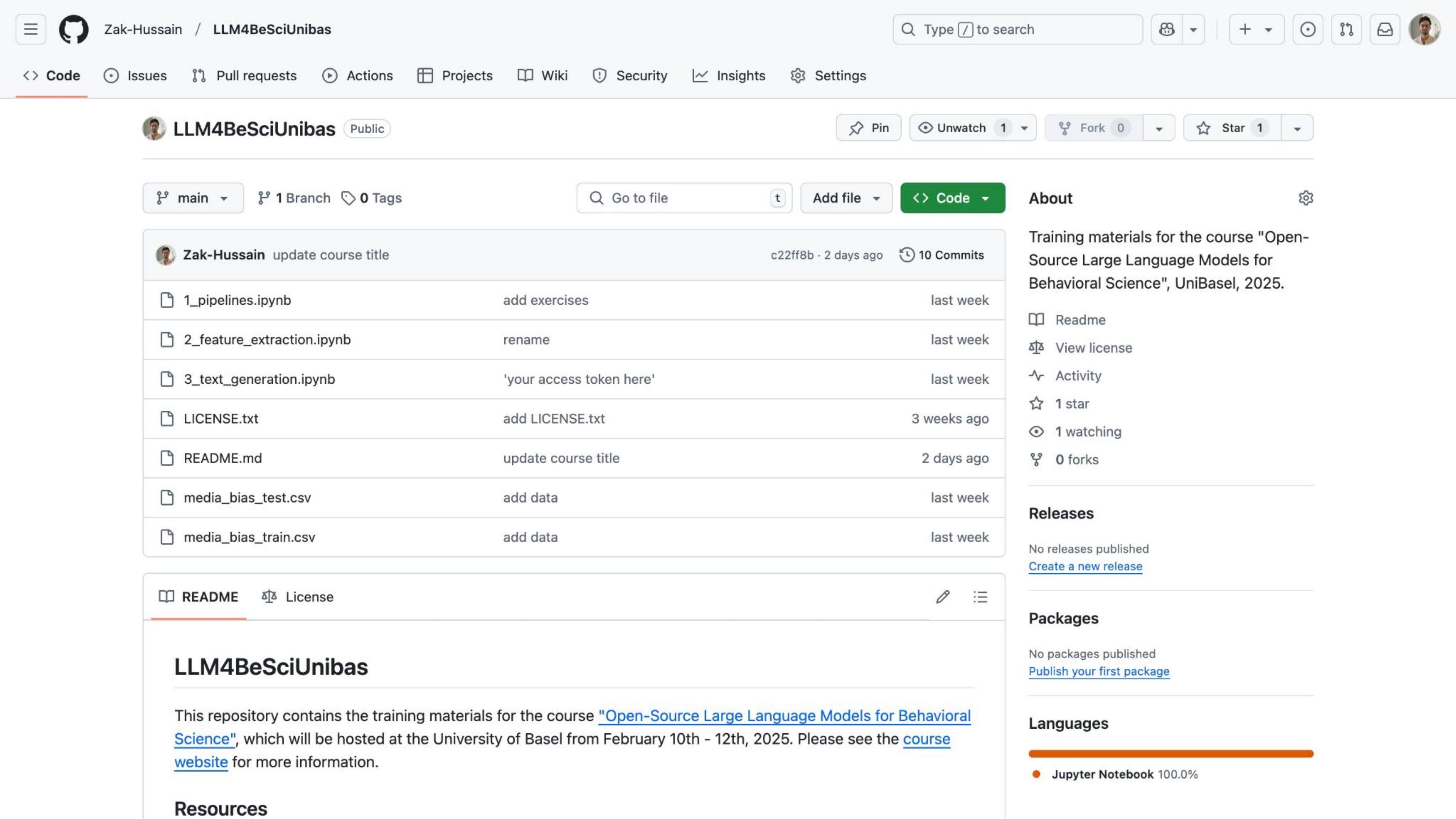
OUR SOFTWARE STACK

python™ + + Google colab

Time to install our stack…

MAX PLANCK INSTITUTE
FOR HUMAN DEVELOPMENT

UNI BASEL


Zak-Hussain / LLM4BeSciUnibas

Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights · Settings

LLM4BeSciUnibas Public

Pin · Unwatch 1 · Fork 0 · Star 1

main · 1 Branch · 0 Tags · Go to file · Add file · Code

Zak-Hussain update course title · c22ff8b · 2 days ago · 10 Commits

1_pipelines.ipynb	add exercises	last week
2_feature_extraction.ipynb	rename	last week
3_text_generation.ipynb	'your access token here'	last week
LICENSE.txt	add LICENSE.txt	3 weeks ago
README.md	update course title	2 days ago
media_bias_test.csv	add data	last week
media_bias_train.csv	add data	last week

README · License

LLM4BeSciUnibas

This repository contains the training materials for the course "Open-Source Large Language Models for Behavioral Science", which will be hosted at the University of Basel from February 10th - 12th, 2025. Please see the course website for more information.

Resources

About

Training materials for the course "Open-Source Large Language Models for Behavioral Science", UniBasel, 2025.

Readme
View license
Activity
1 star
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

● Jupyter Notebook 100.0%