U1.3 Load and explore an audio dataset

```
pip install datasets[audio]
```

```
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets[audio]) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets[audio]) (2024.7.4)
Requirement already satisfied: cffi>=1.0 in /usr/local/lib/python3.10/dist-packages (from soundfile>=0.12.1->datasets[audio]) (1.16.0)
Requirement already satisfied: audioread>=2.1.9 in /usr/local/lib/python3.10/dist-packages (from librosa->datasets[audio]) (3.0.1)
Requirement already satisfied: scipy>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from librosa->datasets[audio]) (1.13.1)
Requirement already satisfied: scikit-learn>=0.20.0 in /usr/local/lib/python3.10/dist-packages (from librosa->datasets[audio]) (1.3.2)
Requirement already satisfied: joblib>=0.14 in /usr/local/lib/python3.10/dist-packages (from librosa->datasets[audio]) (1.4.2)
Requirement already satisfied: decorator>=4.3.0 in /usr/local/lib/python3.10/dist-packages (from librosa->datasets[audio]) (4.4.2)
Requirement already satisfied: numba>=0.51.0 in /usr/local/lib/python3.10/dist-packages (from librosa->datasets[audio]) (0.60.0)
Requirement already satisfied: pooch>=1.1 in /usr/local/lib/python3.10/dist-packages (from librosa->datasets[audio]) (1.8.2)
Requirement already satisfied: soxr>=0.3.2 in /usr/local/lib/python3.10/dist-packages (from librosa->datasets[audio]) (0.4.0)
Requirement already satisfied: lazy-loader>=0.1 in /usr/local/lib/python3.10/dist-packages (from librosa->datasets[audio]) (0.4)
Requirement already satisfied: msgpack>=1.0 in /usr/local/lib/python3.10/dist-packages (from librosa->datasets[audio]) (1.0.8)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets[audio]) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets[audio]) (2024.1)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets[audio]) (2024.1)
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/dist-packages (from cffi>=1.0->soundfile>=0.12.1->datasets[audio]) (2.22)
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/local/lib/python3.10/dist-packages (from numba>=0.51.0->librosa->datasets[audio]) (0.43.0)
Requirement already satisfied: platformdirs>=2.5.0 in /usr/local/lib/python3.10/dist-packages (from pooch>=1.1->librosa->datasets[audio]) (4.2.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas->datasets[audio]) (1.16.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.20.0->librosa->datasets[audio]) (3.5.0)
Downloading dill-0.3.8-py3-none-any.whl (116 kB)
```

```
ibis-framework 8.0.0 requires pyarrow<16,>=2, but you have pyarrow 17.0.0 which is incompatible.
Successfully installed datasets-2.20.0 dill-0.3.8 fsspec-2024.5.0 multiprocess-0.70.16 pyarrow-17.0.0 xxhash-3.4.1
```

Let’s load and explore and audio dataset called MINDS-14, which contains recordings of people asking an e-banking system questions in several languages and dialects.

To load the MINDS-14 dataset, we need to copy the dataset’s identifier on the Hub (PolyAI/minds14) and pass it to the load_dataset function. We’ll also specify that we’re only interested in the Australian subset (en-AU) of the data, and limit it to the training split:

```
from datasets import load_dataset

minds = load_dataset("PolyAI/minds14", name="en-AU", split="train")
minds
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:89: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
Downloading builder script: 100%                                        5.90k/5.90k [00:00<00:00, 5.79kB/s]

Downloading readme: 100%                                               5.29k/5.29k [00:00<00:00, 9.85kB/s]
The repository for PolyAI/minds14 contains custom code which must be executed to correctly load the dataset. You can inspect the repository content at https://hf.co/datasets/PolyAI/minds14.
You can avoid this prompt in future by passing the argument `trust_remote_code=True`.

Do you wish to run the custom code? [y/N] y
Downloading data: 100%                                                 471M/471M [00:14<00:00, 29.7MB/s]

Generating train split:      654/0 [00:00<00:00, 7215.03 examples/s]
Dataset({
    features: ['path', 'audio', 'transcription', 'english_transcription', 'intent_class', 'lang_id'],
    num_rows: 654
})
```

```
example = minds[0]
example
```

```
{'path': '/root/.cache/huggingface/datasets/downloads/extracted/28aa727f91fee90575c34956bab09d1716cfaf460c6afcba86a10f04a7d58b83/en-AU~PAY_BILL/response_4.wav',
 'audio': {'path': '/root/.cache/huggingface/datasets/downloads/extracted/28aa727f91fee90575c34956bab09d1716cfaf460c6afcba86a10f04a7d58b83/en-AU~PAY_BILL/response_4.wav',
  'array': array([ 0.        ,  0.00024414, -0.00024414, ..., -0.00024414,
          0.00024414,  0.0012207 ]),
  'sampling_rate': 8000},
 'transcription': 'I would like to pay my electricity bill using my card can you please assist',
 'english_transcription': 'I would like to pay my electricity bill using my card can you please assist',
 'intent_class': 13,
 'lang_id': 2}
```

You may notice that the audio column contains several features. Here’s what they are:

path: the path to the audio file (*.wav in this case). array: The decoded audio data, represented as a 1-dimensional NumPy array. sampling_rate. The sampling rate of the audio file (8,000 Hz in this example). The intent_class is a classification category of the audio recording. To convert this number into a meaningful string, we can use the int2str() method:

```
id2label = minds.features["intent_class"].int2str
id2label(example["intent_class"])
```

```
'pay_bill'
```

f you plan to train an audio classifier on this subset of data, you may not necessarily need all of the features. For example, the lang_id is going to have the same value for all examples, and won't be useful. The english_transcription will likely duplicate the transcription in this subset, so we can safely remove them.

You can easily remove irrelevant features using 🤗 Datasets' remove_columns method:

```
columns_to_remove = ["lang_id", "english_transcription"]
minds = minds.remove_columns(columns_to_remove)
minds
```

```
Dataset({
    features: ['path', 'audio', 'transcription', 'intent_class'],
    num_rows: 654
})
```

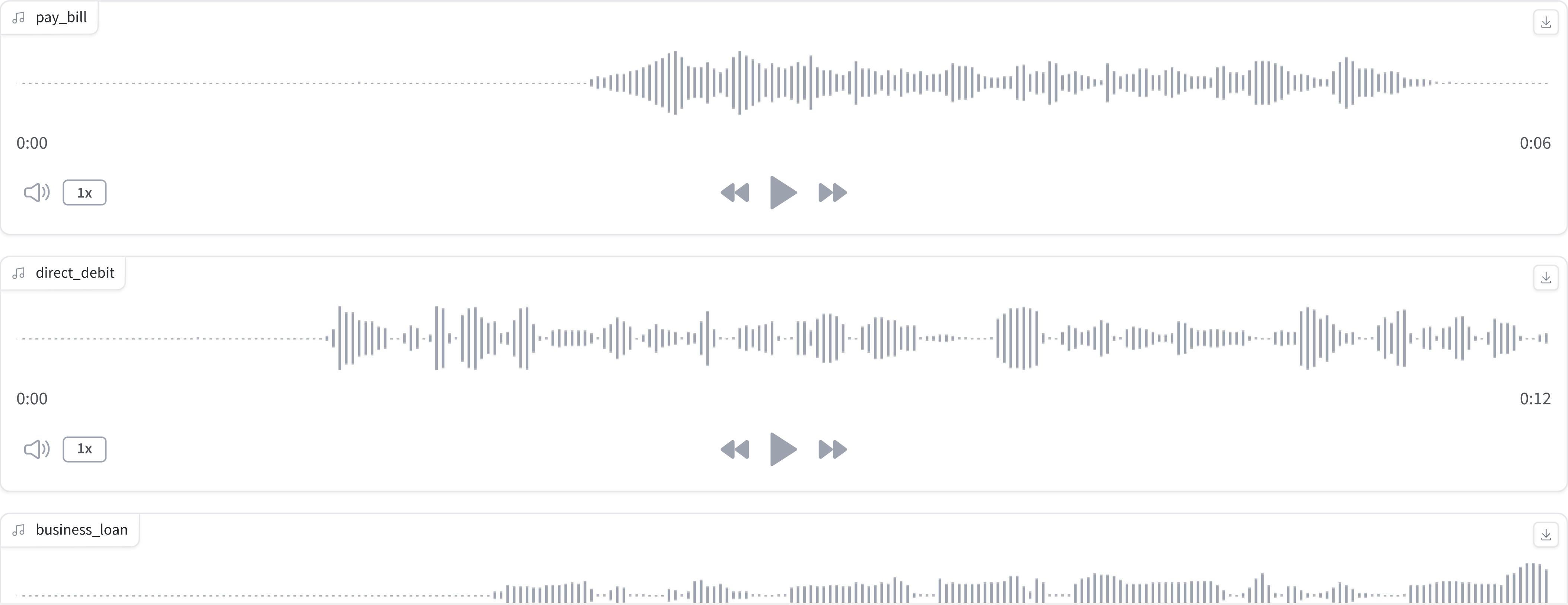
Now that we've loaded and inspected the raw contents of the dataset, let's listen to a few examples! We'll use the Blocks and Audio features from Gradio to decode a few random samples from the dataset:

```
!pip install gradio
```

```
Downloading python_multipart-0.0.9-py3-none-any.whl (22 kB)
Downloading ruff-0.5.6-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (10.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 10.2/10.2 MB 90.4 MB/s eta 0:00:00
Downloading semantic_version-2.10.0-py2.py3-none-any.whl (15 kB)
Downloading uvicorn-0.30.5-py3-none-any.whl (62 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.8/62.8 kB 4.2 MB/s eta 0:00:00
Downloading fastapi-0.112.0-py3-none-any.whl (93 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 93.1/93.1 kB 7.3 MB/s eta 0:00:00
Downloading ffmpy-0.4.0-py3-none-any.whl (5.8 kB)
Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)
Downloading h11-0.14.0-py3-none-any.whl (58 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.3/58.3 kB 4.2 MB/s eta 0:00:00
Downloading starlette-0.37.2-py3-none-any.whl (71 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 71.9/71.9 kB 5.1 MB/s eta 0:00:00
Downloading websockets-12.0-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (130 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 130.2/130.2 kB 8.2 MB/s eta 0:00:00
Installing collected packages: pydub, websockets, tomlkit, semantic-version, ruff, python-multipart, orjson, h11, ffmpy, aiofiles, uvicorn, starlette, httpcore, httpx, fastapi, gradio-client, gradio
  Attempting uninstall: tomlkit
    Found existing installation: tomlkit 0.13.0
    Uninstalling tomlkit-0.13.0:
      Successfully uninstalled tomlkit-0.13.0
Successfully installed aiofiles-23.2.1 fastapi-0.112.0 ffmpy-0.4.0 gradio-4.40.0 gradio-client-1.2.0 h11-0.14.0 httpcore-1.0.5 httpx-0.27.0 orjson-3.10.6 pydub-0.25.1 python-multipart-0.0.9 ruff-0.5.6 semantic-version-2.10.
```

```
import gradio as gr


def generate_audio():
    example = minds.shuffle()[0]
    audio = example["audio"]
    return (
        audio["sampling_rate"],
        audio["array"],
    ), id2label(example["intent_class"])


with gr.Blocks() as demo:
    with gr.Column():
        for _ in range(4):
            audio, label = generate_audio()
            output = gr.Audio(audio, label=label)

demo.launch(debug=True)
```

```
/usr/local/lib/python3.10/dist-packages/gradio/processing_utils.py:574: UserWarning: Trying to convert audio automatically from float64 to 16-bit int format.
  warnings.warn(warning.format(data.dtype))
Setting queue=True in a Colab notebook requires sharing enabled. Setting `share=True` (you can turn this off by setting `share=False` in `launch()` explicitly).

Colab notebook detected. This cell will run indefinitely so that you can see errors and logs. To turn off, set debug=False in launch().
Running on public URL: https://dc2e21a8222476303f.gradio.live

This share link expires in 72 hours. For free permanent hosting and GPU upgrades, run `gradio deploy` from Terminal to deploy to Spaces (https://huggingface.co/spaces)
```



```
Keyboard interruption in main thread... closing server.
Killing tunnel 127.0.0.1:7860 <> https://dc2e21a8222476303f.gradio.live
```

Resampling the audio data

```
from datasets import Audio

minds = minds.cast_column("audio", Audio(sampling_rate=16_000))
```

```
minds[0]
```

```
{'path': '/root/.cache/huggingface/datasets/downloads/extracted/28aa727f91fee90575c34956bab09d1716cfaf460c6afcba86a10f04a7d58b83/en-AU~PAY_BILL/response_4.wav',
 'audio': {'path': '/root/.cache/huggingface/datasets/downloads/extracted/28aa727f91fee90575c34956bab09d1716cfaf460c6afcba86a10f04a7d58b83/en-AU~PAY_BILL/response_4.wav',
  'array': array([2.36119668e-05, 1.92324660e-04, 2.19284790e-04, ...,
         9.40907281e-04, 1.16613181e-03, 7.20883254e-04]),
  'sampling_rate': 16000},
 'transcription': 'I would like to pay my electricity bill using my card can you please assist',
 'intent_class': 13}
```

You may notice that the array values are now also different. This is because we've now got twice the number of amplitude values for every one that we had before.

Filtering the dataset

```
MAX_DURATION_IN_SECONDS = 20.0


def is_audio_length_in_range(input_length):
    return input_length < MAX_DURATION_IN_SECONDS
```

```
!pip install librosa
```

```
Requirement already satisfied: librosa in /usr/local/lib/python3.10/dist-packages (0.10.2.post1)
Requirement already satisfied: audioread>=2.1.9 in /usr/local/lib/python3.10/dist-packages (from librosa) (3.0.1)
Requirement already satisfied: numpy!=1.22.0,!=1.22.1,!=1.22.2,>=1.20.3 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.26.4)
Requirement already satisfied: scipy>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.13.1)
Requirement already satisfied: scikit-learn>=0.20.0 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.3.2)
Requirement already satisfied: joblib>=0.14 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.4.2)
Requirement already satisfied: decorator>=4.3.0 in /usr/local/lib/python3.10/dist-packages (from librosa) (4.4.2)
Requirement already satisfied: numba>=0.51.0 in /usr/local/lib/python3.10/dist-packages (from librosa) (0.60.0)
Requirement already satisfied: soundfile>=0.12.1 in /usr/local/lib/python3.10/dist-packages (from librosa) (0.12.1)
Requirement already satisfied: pooch>=1.1 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.8.2)
Requirement already satisfied: soxr>=0.3.2 in /usr/local/lib/python3.10/dist-packages (from librosa) (0.4.0)
Requirement already satisfied: typing-extensions>=4.1.1 in /usr/local/lib/python3.10/dist-packages (from librosa) (4.12.2)
Requirement already satisfied: lazy-loader>=0.1 in /usr/local/lib/python3.10/dist-packages (from librosa) (0.4)
Requirement already satisfied: msgpack>=1.0 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.0.8)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from lazy-loader>=0.1->librosa) (24.1)
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/local/lib/python3.10/dist-packages (from numba>=0.51.0->librosa) (0.43.0)
Requirement already satisfied: platformdirs>=2.5.0 in /usr/local/lib/python3.10/dist-packages (from pooch>=1.1->librosa) (4.2.2)
Requirement already satisfied: requests>=2.19.0 in /usr/local/lib/python3.10/dist-packages (from pooch>=1.1->librosa) (2.32.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.20.0->librosa) (3.5.0)
Requirement already satisfied: cffi>=1.0 in /usr/local/lib/python3.10/dist-packages (from soundfile>=0.12.1->librosa) (1.16.0)
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/dist-packages (from cffi>=1.0->soundfile>=0.12.1->librosa) (2.22)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->pooch>=1.1->librosa) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->pooch>=1.1->librosa) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->pooch>=1.1->librosa) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->pooch>=1.1->librosa) (2024.7.4)
```

```
import librosa
# use librosa to get example's duration from the audio file
new_column = [librosa.get_duration(path=x) for x in minds["path"]]
minds = minds.add_column("duration", new_column)

# use 🤗 Datasets' `filter` method to apply the filtering function
minds = minds.filter(is_audio_length_in_range, input_columns=["duration"])

# remove the temporary helper column
minds = minds.remove_columns(["duration"])
minds
```

```
Filter: 100%                                        654/654 [00:00<00:00, 8187.43 examples/s]

Dataset({
    features: ['path', 'audio', 'transcription', 'intent_class'],
    num_rows: 624
})
```

We can verify that dataset has been filtered down from 654 examples to 624.

Pre-processing audio data

One of the most challenging aspects of working with audio datasets is preparing the data in the right format for model training. As you saw, the raw audio data comes as an array of sample values. However, pre-trained models, whether you use them for inference, or want to fine-tune them for your task, expect the raw data to be converted into input features. The requirements for the input features may vary from one model to another — they depend on the model's architecture, and the data it was pre-trained with. The good news is, for every supported audio model, 🤗 Transformers offer a feature extractor class that can convert raw audio data into the input features the model expects.

So what does a feature extractor do with the raw audio data? Let's take a look at Whisper's feature extractor to understand some common feature extraction transformations. Whisper is a pre-trained model for automatic speech recognition (ASR) published in September 2022 by Alec Radford et al. from OpenAI.

First, the Whisper feature extractor pads/truncates a batch of audio examples such that all examples have an input length of 30s. Examples shorter than this are padded to 30s by appending zeros to the end of the sequence (zeros in an audio signal correspond to no signal or silence). Examples longer than 30s are truncated to 30s. Since all elements in the batch are padded/truncated to a maximum length in the input space, there is no need for an attention mask. Whisper is unique in this regard, most other audio models require an attention mask that details where sequences have been padded, and thus where they should be ignored in the self-attention mechanism. Whisper is trained to operate without an attention mask and infer directly from the speech signals where to ignore the inputs.

The second operation that the Whisper feature extractor performs is converting the padded audio arrays to log-mel spectrograms. As you recall, these spectrograms describe how the frequencies of a signal change over time, expressed on the mel scale and measured in decibels (the log part) to make the frequencies and amplitudes more representative of human hearing.

All these transformations can be applied to your raw audio data with a couple of lines of code. Let's go ahead and load the feature extractor from the pre-trained Whisper checkpoint to have ready for our audio data:

```
from transformers import WhisperFeatureExtractor

feature_extractor = WhisperFeatureExtractor.from_pretrained("openai/whisper-small")
```

```
preprocessor_config.json:  100%                                    185k/185k  [00:00<00:00,  6.76MB/s]
```

```
def prepare_dataset(example):
    audio = example["audio"]
    features = feature_extractor(
        audio["array"], sampling_rate=audio["sampling_rate"], padding=True
    )
    return features


minds = minds.map(prepare_dataset)
minds
```

```
Map:  100%                                    624/624  [01:57<00:00,   2.83s/  examples]
Dataset({
    features: ['path', 'audio', 'transcription', 'intent_class', 'input_features'],
    num_rows: 624
})
```

As easy as that, we now have log-mel spectrograms as input_features in the dataset.

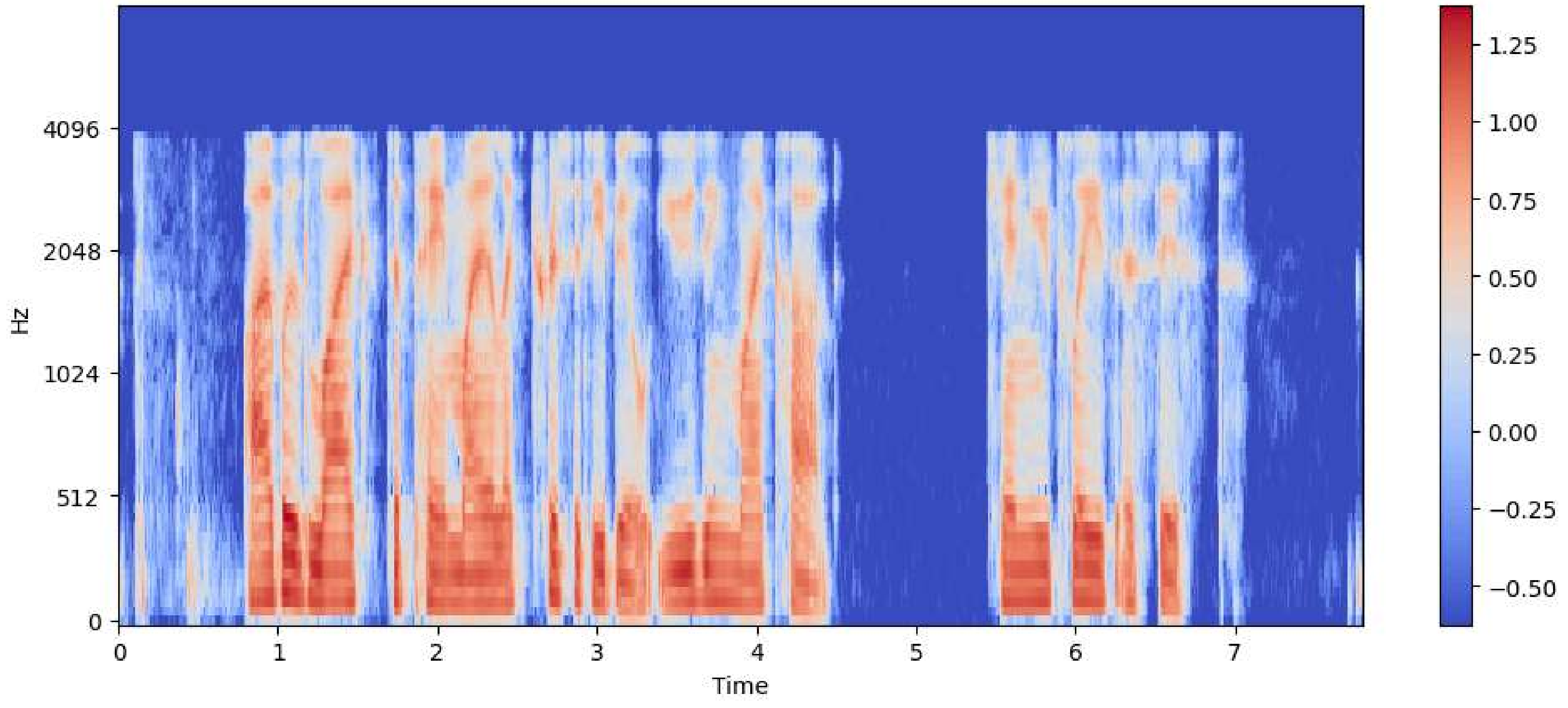
Let's visualize it for one of the examples in the minds dataset:

```
!pip install matplotlib
```

```
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.2.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.53.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.5)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.26.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
```

```
import numpy as np
import matplotlib.pyplot as plt
example = minds[0]
input_features = example["input_features"]

plt.figure().set_figwidth(12)
librosa.display.specshow(
    np.asarray(input_features[0]),
    x_axis="time",
    y_axis="mel",
    sr=feature_extractor.sampling_rate,
    hop_length=feature_extractor.hop_length,
)
plt.colorbar()
```

```
<matplotlib.colorbar.Colorbar at 0x7b312c92e3e0>
```



Kodlamaya başlayın veya yapay zeka ile kod oluşturun.

Kodlamaya başlayın veya yapay zeka ile kod oluşturun.