Process audio data

This guide shows specific methods for processing audio datasets. Learn how to:

- Resample the sampling rate.
- Use map() with audio datasets.

For a guide on how to process any type of dataset, take a look at the general process guide.

Cast

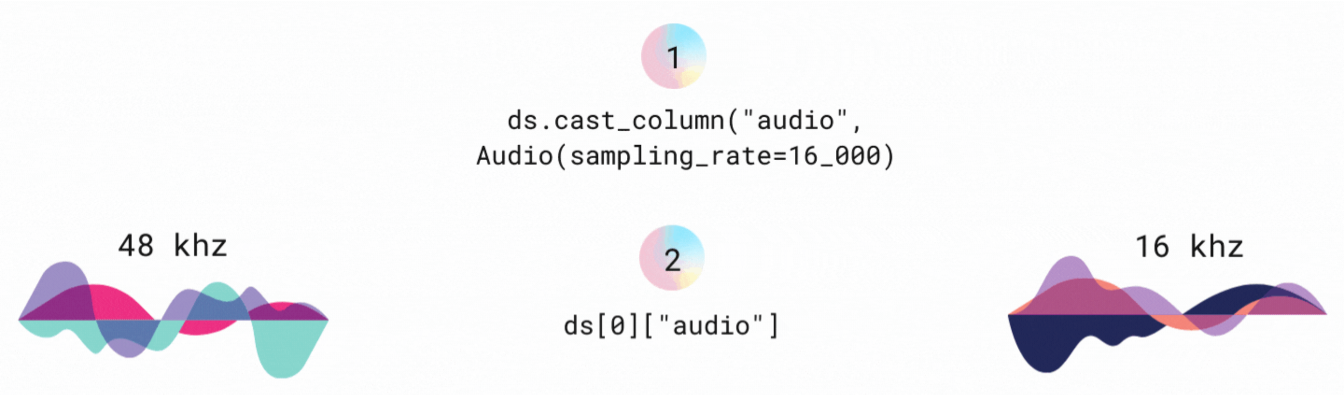
The cast_column() function is used to cast a column to another feature to be decoded. When you use this function with the Audio feature, you can resample the sampling rate:

```
>>> from datasets import load_dataset, Audio

>>> dataset = load_dataset("PolyAI/minds14", "en-US", split="train")
>>> dataset = dataset.cast_column("audio", Audio(sampling_rate=16000))
```

Audio files are decoded and resampled on-the-fly, so the next time you access an example, the audio file is resampled to 16kHz:

```
>>> audio = dataset[0]["audio"]
<datasets.features._torchcodec.AudioDecoder object at 0x11642b6a0>
>>> audio = audio_dataset[0]["audio"]
>>> samples = audio.get_all_samples()
>>> samples.data
tensor([[ 0.0000e+00,  0.0000e+00,  0.0000e+00,  ...,  2.3447e-06,
         -1.9127e-04, -5.3330e-05]]
>>> samples.sample_rate
16000
```



Map

The map() function helps preprocess your entire dataset at once. Depending on the type of model you're working with, you'll need to either load a feature extractor or a processor.

- For pretrained speech recognition models, load a feature extractor and tokenizer and combine them in a `processor`:

```
>>> from transformers import AutoTokenizer, AutoFeatureExtractor, AutoProcessor

>>> model_checkpoint = "facebook/wav2vec2-large-xlsr-53"
# after defining a vocab.json file you can instantiate a tokenizer object:
>>> tokenizer = AutoTokenizer("./vocab.json", unk_token="[UNK]", pad_token="[PAD]", word_delim
>>> feature_extractor = AutoFeatureExtractor.from_pretrained(model_checkpoint)
>>> processor = AutoProcessor.from_pretrained(feature_extractor=feature_extractor, tokenizer=t
```

- For fine-tuned speech recognition models, you only need to load a `processor`:

```
>>> from transformers import AutoProcessor

>>> processor = AutoProcessor.from_pretrained("facebook/wav2vec2-base-960h")
```

When you use map() with your preprocessing function, include the `audio` column to ensure you're actually resampling the audio data:

```
>>> def prepare_dataset(batch):
...     audio = batch["audio"]
...     batch["input_values"] = processor(audio.get_all_samples().data, sampling_rate=audio["sampl
...     batch["input_length"] = len(batch["input_values"])
...     with processor.as_target_processor():
...         batch["labels"] = processor(batch["sentence"]).input_ids
...     return batch
>>> dataset = dataset.map(prepare_dataset, remove_columns=dataset.column_names)
```