Introduction to audio data in Python

SPOKEN LANGUAGE PROCESSING IN PYTHON

Daniel Bourke

Machine Learning Engineer/YouTube Creator

datacamp

Dealing with audio files in Python

- Different kinds all of audio files
  - mp3
  - wav
  - m4a
  - flac
- Digital sounds measured in frequency (kHz)
  - 1 kHz = 1000 pieces of information per second

Frequency examples

- Streaming songs have a frequency of 32 kHz
- Audiobooks and spoken language are between 8 and 16 kHz
- We can't see audio files so we have to transform them first

```
import wave
```

Opening an audio file in Python

- Audio file saved as `good-morning.wav`

```
# Import audio file as wave object
good_morning = wave.open("good-morning.wav", "r")
```

```
# Convert wave object to bytes
good_morning_soundwave = good_morning.readframes(-1)
```

```
# View the wav file in byte form
good_morning_soundwave
```

```
b'\xfd\xff\xfb\xff\xf8\xff\xf8\xff\xf7\...
```

Working with audio is different

- Have to convert the audio to something useful
- Small sample of audio = large amount of information

Let's practice!

SPOKEN LANGUAGE PROCESSING IN PYTHON

Converting sound wave bytes to integers

SPOKEN LANGUAGE PROCESSING IN PYTHON

Daniel Bourke

Machine Learning Engineer/YouTube Creator

datacamp

Converting bytes to integers

- Can't use bytes
- Convert bytes to integers using numpy

```
import numpy as np
# Convert soundwave_gm from bytes to integers
signal_gm = np.frombuffer(soundwave_gm, dtype='int16')
# Show the first 10 items
signal_gm[:10]
```

```
array([ -3,  -5,  -8,  -8,  -9, -13,  -8, -10,  -9, -11], dtype=int16)
```

Finding the frame rate

- Frequency (Hz) = length of wave object array/duration of audio file (seconds)

```
# Get the frame rate
framerate_gm = good_morning.getframerate()
# Show the frame rate
framerate_gm
```

```
48,000
```

- Duration of audio file (seconds) = length of wave object array/frequency (Hz)

Finding sound wave timestamps

```
# Return evenly spaced values between start and stop
np.linspace(start=1, stop=10, num=10)
```

```
array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
```

```
# Get the timestamps of the good morning sound wave
time_gm = np.linspace(start=0,
                      stop=len(soundwave_gm)/framerate_gm,
                      num=len(soundwave_gm))
```

Finding sound wave timestamps

```
# View first 10 time stamps of good morning sound wave
time_gm[:10]
```

```
array([0.00000000e+00, 2.08334167e-05, 4.16668333e-05, 6.25002500e-05,
       8.33336667e-05, 1.04167083e-04, 1.25000500e-04, 1.45833917e-04,
       1.66667333e-04, 1.87500750e-04])
```

Let's practice!

SPOKEN LANGUAGE PROCESSING IN PYTHON

Visualizing sound waves

SPOKEN LANGUAGE PROCESSING IN PYTHON

Daniel Bourke

Machine Learning Engineer/YouTube Creator

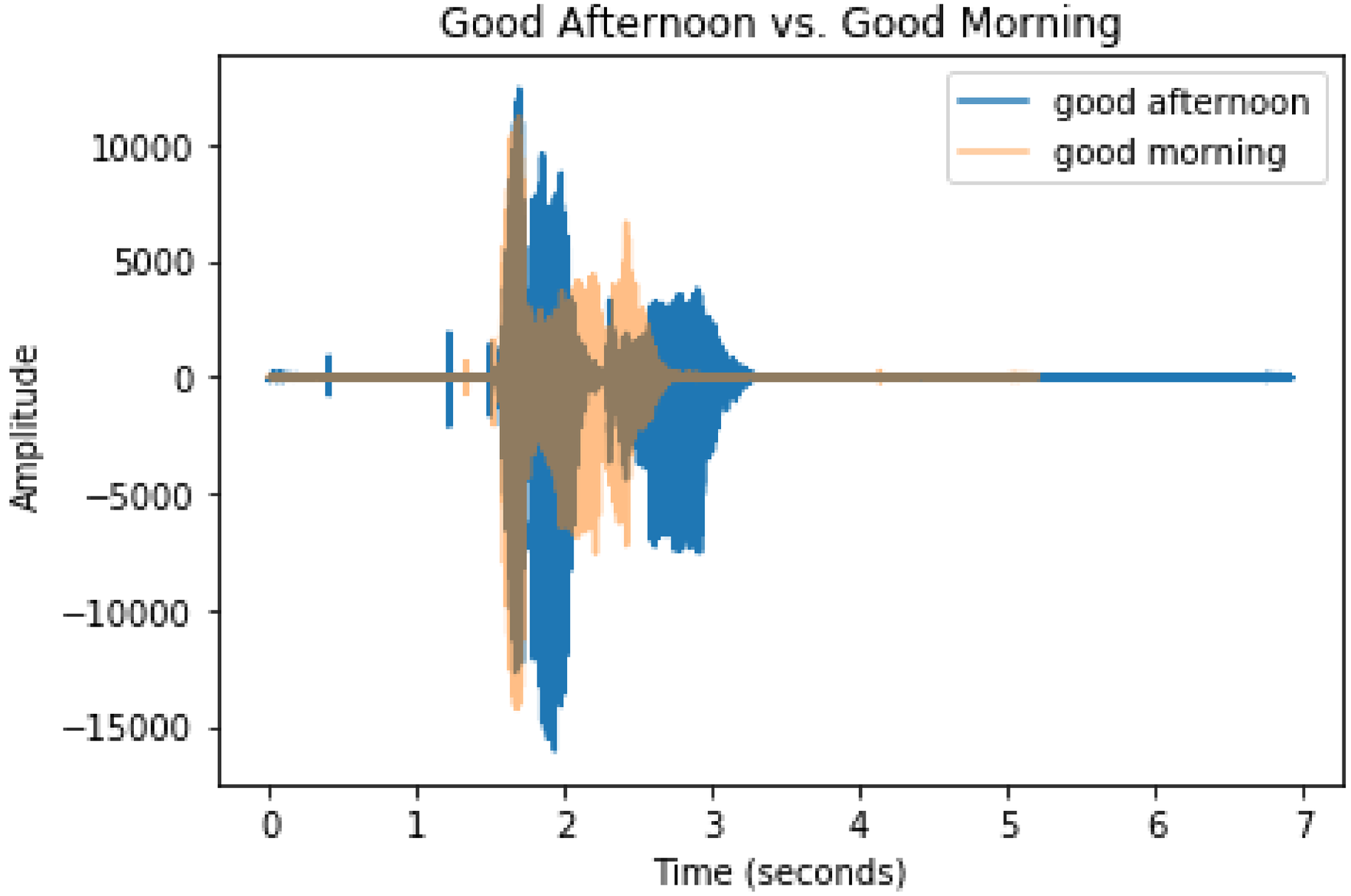
datacamp

Adding another sound wave

- New audio file: `good_afternoon.wav`
- Both are 48 kHz
- Same data transformations to *all* audio files

Setting up a plot

```
import matplotlib.pyplot as plt
# Initialize figure and setup title
plt.title("Good Afternoon vs. Good Morning")
# x and y axis labels
plt.xlabel("Time (seconds)")
plt.ylabel("Amplitude")
# Add good morning and good afternoon values
plt.plot(time_ga, soundwave_ga, label ="Good Afternoon")
plt.plot(time_gm, soundwave_gm, label="Good Morning",
         alpha=0.5)
# Create a legend and show our plot
plt.legend()
plt.show()
```


Good Afternoon vs. Good Morning
good afternoon
good morning
10000
5000
0
−5000
−10000
−15000
Amplitude
0
1
2
3
4
5
6
7
Time (seconds)

Time to visualize!

SPOKEN LANGUAGE PROCESSING IN PYTHON