Understanding Speech Data using Python

Bindeshwar Singh Kushwaha
Postnetwork Academy

Understanding Speech Data

- **What is Speech Data?**

Understanding Speech Data

- **What is Speech Data?**
  - Audio recordings containing human voice signals

Understanding Speech Data

- **What is Speech Data?**
  - Audio recordings containing human voice signals
  - Represented digitally as a sequence of amplitude values over time

Understanding Speech Data

- **What is Speech Data?**
  - Audio recordings containing human voice signals
  - Represented digitally as a sequence of amplitude values over time
- **Two Main Forms:**

Understanding Speech Data

- **What is Speech Data?**
  - Audio recordings containing human voice signals
  - Represented digitally as a sequence of amplitude values over time
- **Two Main Forms:**
  - Time-Domain: Waveform showing amplitude over time

Understanding Speech Data

- **What is Speech Data?**
  - Audio recordings containing human voice signals
  - Represented digitally as a sequence of amplitude values over time
- **Two Main Forms:**
  - Time-Domain: Waveform showing amplitude over time
  - Frequency-Domain: Spectrogram showing energy distribution across frequencies

Understanding Speech Data

- **What is Speech Data?**
  - Audio recordings containing human voice signals
  - Represented digitally as a sequence of amplitude values over time
- **Two Main Forms:**
  - Time-Domain: Waveform showing amplitude over time
  - Frequency-Domain: Spectrogram showing energy distribution across frequencies
- **Why Process Speech Data?**

Understanding Speech Data

- **What is Speech Data?**
  - Audio recordings containing human voice signals
  - Represented digitally as a sequence of amplitude values over time
- **Two Main Forms:**
  - Time-Domain: Waveform showing amplitude over time
  - Frequency-Domain: Spectrogram showing energy distribution across frequencies
- **Why Process Speech Data?**
  - Speech recognition (transcription)

Understanding Speech Data

- **What is Speech Data?**
  - Audio recordings containing human voice signals
  - Represented digitally as a sequence of amplitude values over time
- **Two Main Forms:**
  - Time-Domain: Waveform showing amplitude over time
  - Frequency-Domain: Spectrogram showing energy distribution across frequencies
- **Why Process Speech Data?**
  - Speech recognition (transcription)
  - Command detection

Understanding Speech Data

- **What is Speech Data?**
  - Audio recordings containing human voice signals
  - Represented digitally as a sequence of amplitude values over time
- **Two Main Forms:**
  - Time-Domain: Waveform showing amplitude over time
  - Frequency-Domain: Spectrogram showing energy distribution across frequencies
- **Why Process Speech Data?**
  - Speech recognition (transcription)
  - Command detection
  - Speaker identification and emotion detection

Loading and Displaying Speech Data in Librosa

Python Example

```
import librosa
import librosa.display
import matplotlib.pyplot as plt
# Load the speech audio file
y, sr = librosa.load("stop.wav", sr=None)
# Display basic information
print(f"Sample rate: {sr} Hz")
print(f"Audio length: {len(y)/sr:.2f} seconds")
# Plot the waveform
plt.figure(figsize=(8, 3))
librosa.display.waveshow(y, sr=sr)
plt.title("Speech Waveform")
plt.xlabel("Time (seconds)")
plt.ylabel("Amplitude")
plt.tight_layout()
plt.show()
```

Loading and Inspecting Speech Data

- **Import Libraries**
  - `librosa` for audio processing

Loading and Inspecting Speech Data

- **Import Libraries**
  - `librosa` for audio processing
  - `librosa.display` for visualization

Loading and Inspecting Speech Data

- **Import Libraries**
  - `librosa` for audio processing
  - `librosa.display` for visualization
  - `matplotlib.pyplot` for plotting

Loading and Inspecting Speech Data

- **Import Libraries**
  - `librosa` for audio processing
  - `librosa.display` for visualization
  - `matplotlib.pyplot` for plotting
- **Load the Audio File**
  - `y, sr = librosa.load("stop.wav", sr=None)`

Loading and Inspecting Speech Data

- **Import Libraries**
  - `librosa` for audio processing
  - `librosa.display` for visualization
  - `matplotlib.pyplot` for plotting
- **Load the Audio File**
  - `y, sr = librosa.load("stop.wav", sr=None)`
  - Loads waveform data into `y` (NumPy array)

Loading and Inspecting Speech Data

- **Import Libraries**
  - `librosa` for audio processing
  - `librosa.display` for visualization
  - `matplotlib.pyplot` for plotting
- **Load the Audio File**
  - `y, sr = librosa.load("stop.wav", sr=None)`
  - Loads waveform data into `y` (NumPy array)
  - `sr` stores the sampling rate in Hz

Loading and Inspecting Speech Data

- **Import Libraries**
  - `librosa` for audio processing
  - `librosa.display` for visualization
  - `matplotlib.pyplot` for plotting
- **Load the Audio File**
  - `y, sr = librosa.load("stop.wav", sr=None)`
  - Loads waveform data into `y` (NumPy array)
  - `sr` stores the sampling rate in Hz
- **Display Basic Info**
  - Prints sample rate

Loading and Inspecting Speech Data

- **Import Libraries**
  - `librosa` for audio processing
  - `librosa.display` for visualization
  - `matplotlib.pyplot` for plotting
- **Load the Audio File**
  - `y, sr = librosa.load("stop.wav", sr=None)`
  - Loads waveform data into `y` (NumPy array)
  - `sr` stores the sampling rate in Hz
- **Display Basic Info**
  - Prints sample rate
  - Calculates duration as `len(y)/sr` seconds

Plotting the Speech Waveform

- **Set Figure Size**
  - `plt.figure(figsize=(8, 3))` makes the plot wide and short

Plotting the Speech Waveform

- **Set Figure Size**
  - `plt.figure(figsize=(8, 3))` makes the plot wide and short
- **Plot Waveform**
  - `librosa.display.waveshow(y, sr=sr)` draws amplitude vs time

Plotting the Speech Waveform

- **Set Figure Size**
  - `plt.figure(figsize=(8, 3))` makes the plot wide and short
- **Plot Waveform**
  - `librosa.display.waveshow(y, sr=sr)` draws amplitude vs time
- **Add Labels and Title**
  - `plt.title("Speech Waveform")`

Plotting the Speech Waveform

- **Set Figure Size**
  - `plt.figure(figsize=(8, 3))` makes the plot wide and short
- **Plot Waveform**
  - `librosa.display.waveshow(y, sr=sr)` draws amplitude vs time
- **Add Labels and Title**
  - `plt.title("Speech Waveform")`
  - X-axis: Time (seconds)

Plotting the Speech Waveform

- **Set Figure Size**
  - `plt.figure(figsize=(8, 3))` makes the plot wide and short
- **Plot Waveform**
  - `librosa.display.waveshow(y, sr=sr)` draws amplitude vs time
- **Add Labels and Title**
  - `plt.title("Speech Waveform")`
  - X-axis: Time (seconds)
  - Y-axis: Amplitude

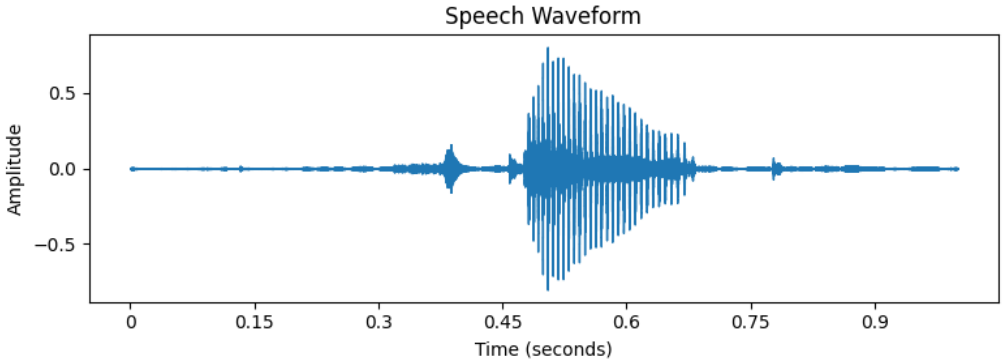
Plotting the Speech Waveform

- **Set Figure Size**
  - `plt.figure(figsize=(8, 3))` makes the plot wide and short
- **Plot Waveform**
  - `librosa.display.waveshow(y, sr=sr)` draws amplitude vs time
- **Add Labels and Title**
  - `plt.title("Speech Waveform")`
  - X-axis: Time (seconds)
  - Y-axis: Amplitude
- **Optimize Layout and Show**
  - `plt.tight_layout()` prevents label overlap

Plotting the Speech Waveform

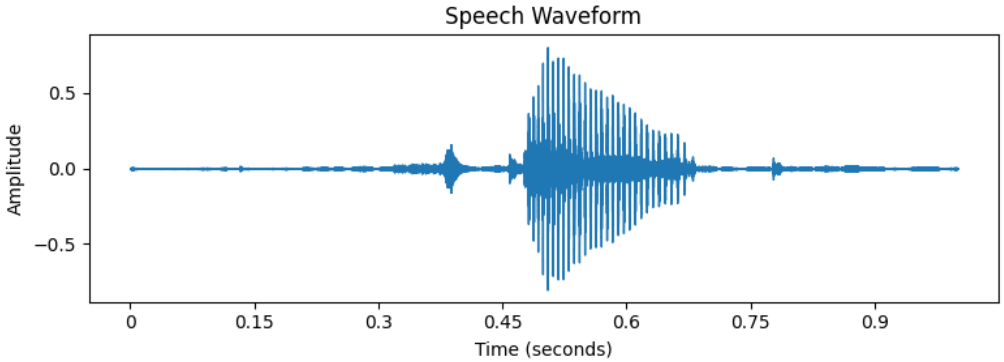
- **Set Figure Size**
  - `plt.figure(figsize=(8, 3))` makes the plot wide and short
- **Plot Waveform**
  - `librosa.display.waveshow(y, sr=sr)` draws amplitude vs time
- **Add Labels and Title**
  - `plt.title("Speech Waveform")`
  - X-axis: Time (seconds)
  - Y-axis: Amplitude
- **Optimize Layout and Show**
  - `plt.tight_layout()` prevents label overlap
  - `plt.show()` displays the final plot

Output: Speech Waveform Visualization



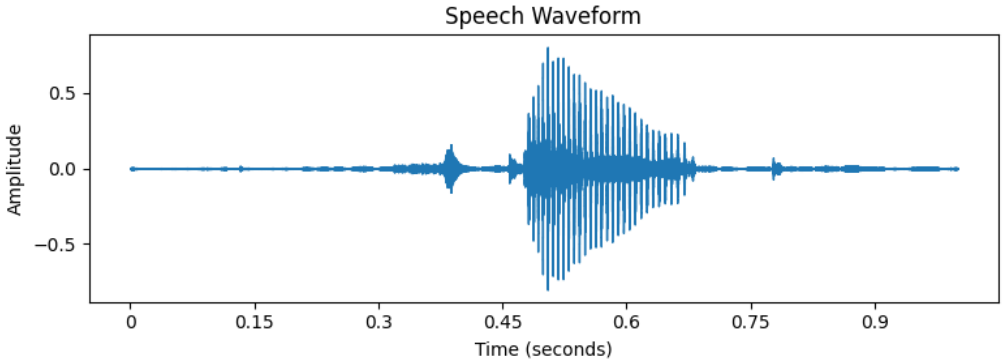
- X-axis: Time (seconds)

Output: Speech Waveform Visualization



- X-axis: Time (seconds)
- Y-axis: Amplitude

Output: Speech Waveform Visualization



- X-axis: Time (seconds)
- Y-axis: Amplitude
- Peaks and troughs indicate speech patterns

Thank You!