# Speech Processing Assignment

GitHub Link : https://github.com/Toji339/SpeechProcessing

## Dataset Description :

- This LJ Speech dataset has been taken from the Kaggle.
- It contains 13,100 short audio clips of a single speaker
- These clips vary in length from 1 to 10 sec and have a total length of around 24 hours.

## Objective :

- 
  - To gain hands on experience with regarding speech signals.
  - Observe how the speech signals are changing while amplification, de-amplification,Up sampling and Down Sampling.
  - To get familiar with libraries like librosa, scipy , soundfile and simpleaudio.

## Code :

Python
```python


import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import librosa
import librosa.display

filename = 'LJ038-0014.wav'

"""# Task 1"""

plt.figure(figsize = (14,7))
data, sample_rate = librosa.load(filename)
librosa.display.waveshow(data, sr = sample_rate)
plt.title("Waveform of Audio File")
plt.xlabel("Time (s)")
plt.ylabel("Amplitude")
plt.show()

"""## Let's hear"""

```

```python
from IPython.display import Audio

data, sample_rate = librosa.load(filename)
plt.figure(figsize = (14,7))
librosa.display.waveshow(data,sr = sample_rate)
Audio(filename)

"""# Task 2"""

# Extracting first 2 seconds of the signal

import soundfile as sf
data, sr = sf.read(filename)
duration = 2
num_samples = int(duration * sr)
first_two_seconds = data[:num_samples]
output_filename = 'first_2sec.wav'
sf.write(output_filename, first_two_seconds, sr)
```

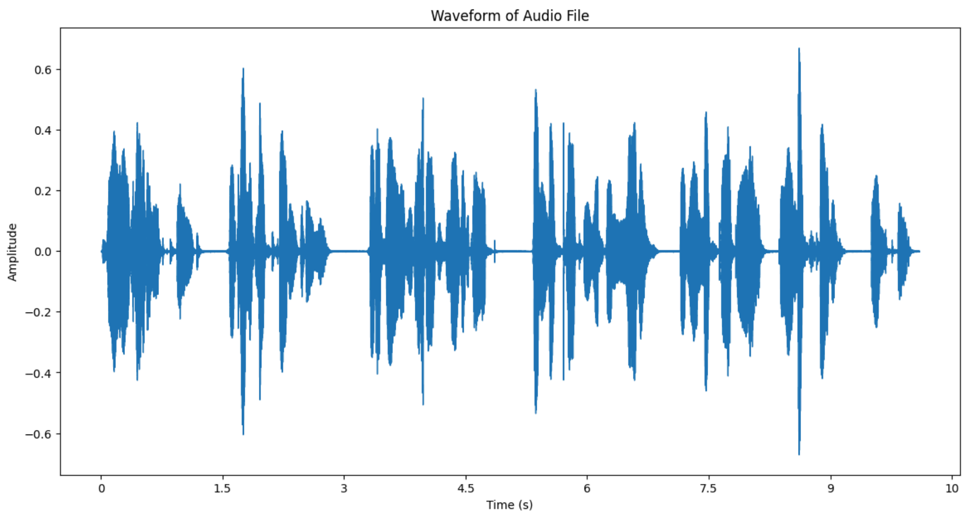


Figure 1: Speech Signal .

## Normalizing :

Python

```python
# Normalizing the extracted values

first_2_sec_normalized = librosa.util.normalize(first_two_seconds)

plt.figure(figsize = (14,7))
librosa.display.waveshow(first_2_sec_normalized, sr = sample_rate)
plt.title("Waveform of first 2 sec")
plt.xlabel("Time (s)")
```

```
plt.ylabel("Amplitude")
plt.show()
```

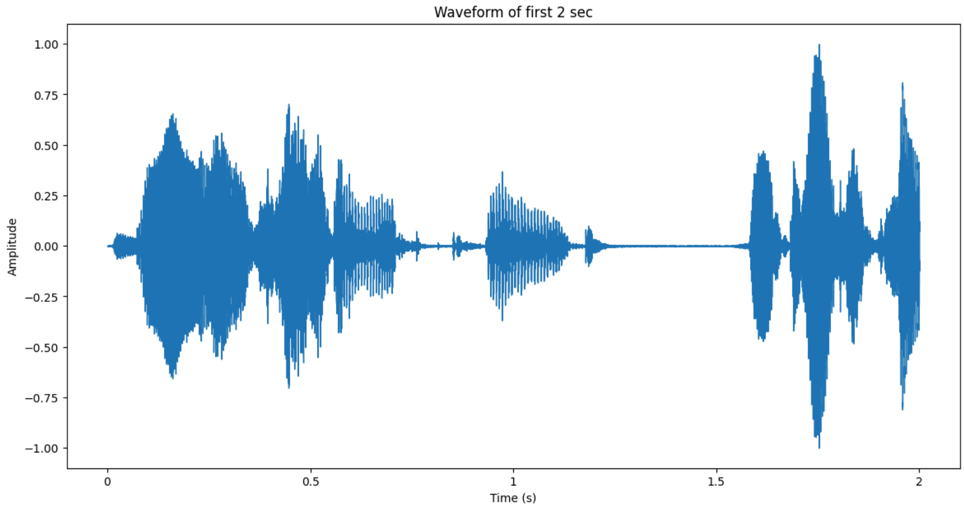


Figure 2: Speech Signal After Normalizing .

## Amplification :

```python

"""# Amplification"""

amplification_value = 2
first_2_sec_amplified = first_2_sec_normalized * amplification_value


plt.figure(figsize = (14,7))
librosa.display.waveshow(first_2_sec_amplified, sr = sample_rate)
plt.title("Waveform of Amplified first 2 sec")
plt.xlabel("Time (s)")
plt.ylabel("Amplitude")
plt.show()
```

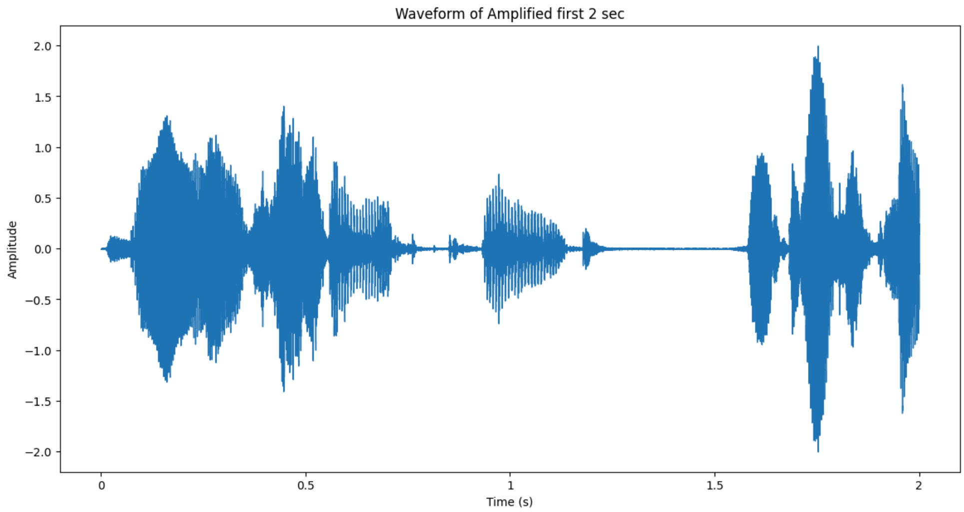


Figure 3: Amplified Speech Signal .

## De-Amplification :

```python

"""# Deamplification

"""

de_amplification_value = 0.5
first_2_sec_deamplified = first_2_sec_amplified * de_amplification_value

plt.figure(figsize=(14, 7))
librosa.display.waveshow(first_2_sec_deamplified, sr=sample_rate)
plt.title("Waveform of De-amplified First 2 Seconds")
plt.xlabel("Time (s)")
plt.ylabel("Amplitude")
plt.show()
```

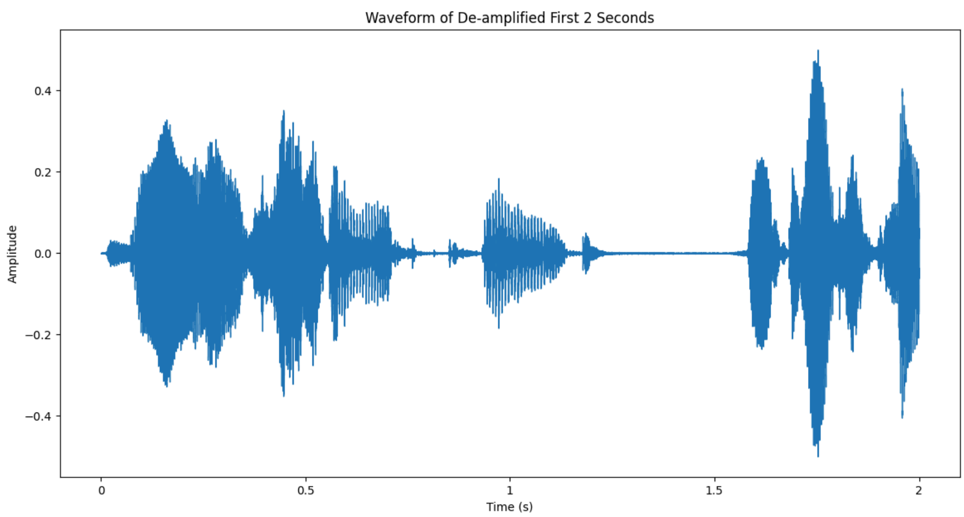


Figure 4: De-amplified Speech Signal .

## Up sampling :

Python

```python

"""# UpSampling"""

up_sample_rate = sample_rate * 2
first_2_sec_up_sampled = librosa.resample(first_2_sec_normalized, orig_sr =
sample_rate, target_sr = up_sample_rate)

plt.figure(figsize=(14, 7))
librosa.display.waveshow(first_2_sec_up_sampled, sr=up_sample_rate)
plt.title("Waveform of Up-sampled First 2 Seconds")
plt.xlabel("Time (s)")
plt.ylabel("Amplitude")
plt.show()
```

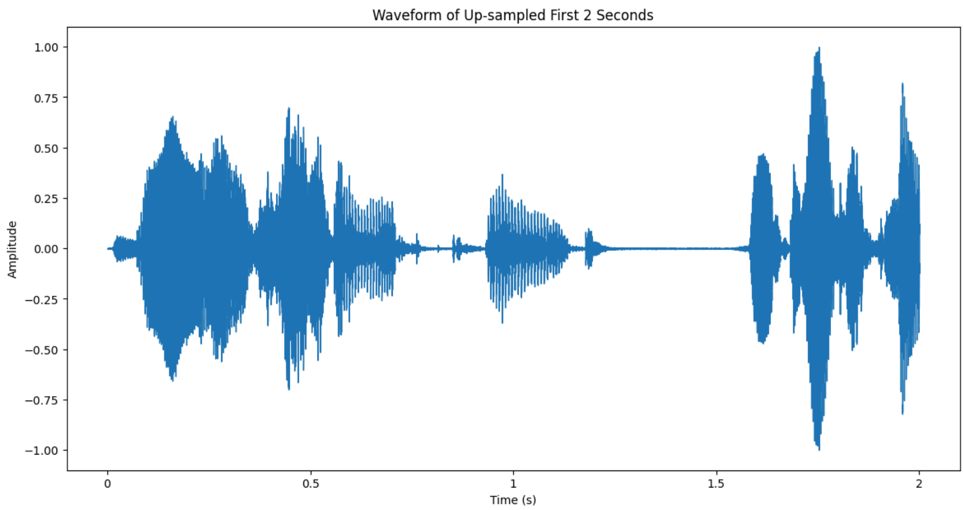


Figure 5: Upsampled Speech Signal .

## Down Sampling :

Python

```python
"""# DownSampling"""

down_sample_rate = sample_rate // 2
first_2_sec_down_sampled = librosa.resample(first_2_sec_normalized, orig_sr =
sample_rate, target_sr = down_sample_rate)

plt.figure(figsize=(14, 7))
librosa.display.waveshow(first_2_sec_down_sampled, sr=down_sample_rate)
plt.title("Waveform of Down-sampled First 2 Seconds")
plt.xlabel("Time (s)")
plt.ylabel("Amplitude")
plt.show()
```

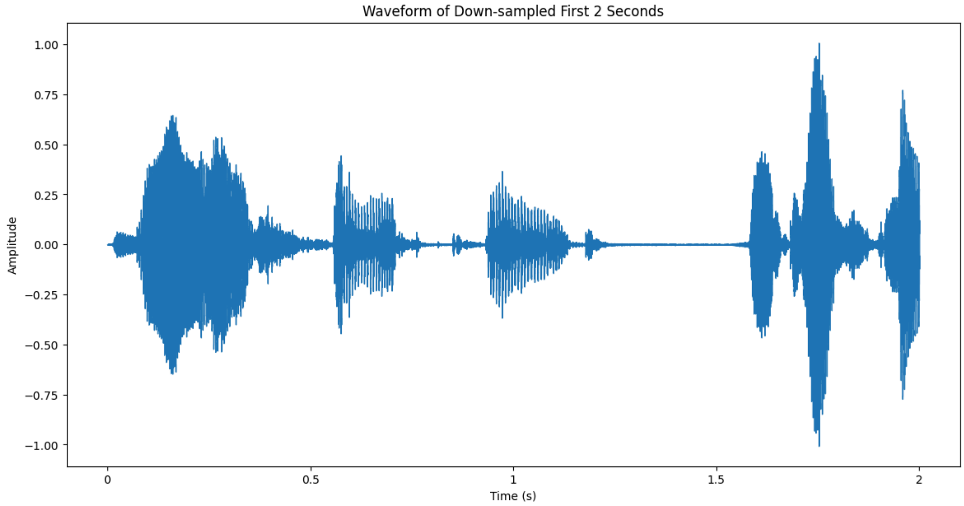


Figure 6: Down-sampled Speech Signal.

## Conclusion

- In the original signal the amplitude was ranging from $-0.6$ to $0.6$ but after normalizing the values around $-0.9$ to $0.9$ , actually there is some improvement because already the values have been in the range of $-1$ to $1$ , now after normalizing the values increased a bit.
- After taking only the first 2sec of the signal , when we amplified it the amplitude range changed into approx $-1.9$ to $1.9$ , because we amplified the signal by a factor of 2.
- When we de-amplified the signal the values range changed into approx $-0.3$ to $0.3$ , we have de-amplified the normalized signal by a factor of $1 / 2$.
- When the sample rate is doubled, the number of samples in the signal increases, but the original frequency content remains unchanged.we can smooth the transition and reconstruct the missing values by passing it through a low pass filter.
- When the sample rate is reduced (down-sampling), the number of samples in the signal decreases, but the original frequency content remains the same.