BL.EN.U4AIE21032
B.CHAITANYA KIRAN

Lab Assignment – 1: Introduction to Speech Processing

Title of the Experiment:

Introduction to Speech Processing and Basic Signal Operations

Objective:

- To record or use a speech signal and determine its sampling rate and bit depth.
- To visualize the speech signal in the time domain.
- To perform slicing and normalization.
- To apply amplification, attenuation, up-sampling, and down-sampling.
- To visually identify voiced, unvoiced, and silence regions.

Dataset Description:

The dataset used in this experiment is taken from the **LJSpeech Dataset**, which contains short speech recordings in WAV format. One audio sample was selected from the dataset and processed using Python in Google Colab/Kaggle. The audio is stored in WAV format with standard 16-bit PCM encoding, which is commonly used in speech processing tasks.

Tools and Platform Used:

- **Platform: Kaggle Notebook**
- **Programming Language: Python**
- **Libraries Used: NumPy, Matplotlib, Librosa, SoundFile, SciPy**

Code:

```
import numpy as np

import pandas as pd

import os

import matplotlib.pyplot as plt

import librosa
```

BL.EN.U4AIE21032
B.CHAITANYA KIRAN

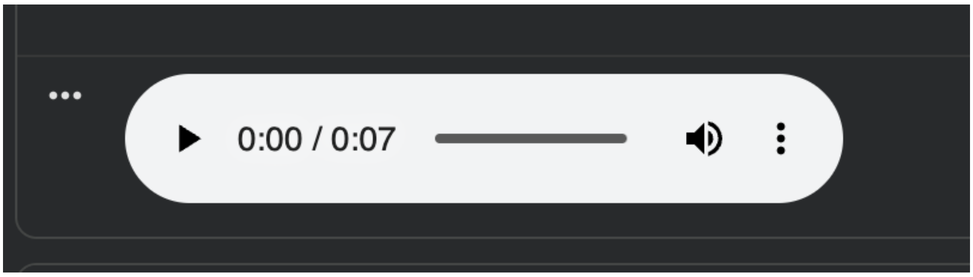
```
import soundfile as sf

from scipy.signal import resample

csv_path = "/kaggle/input/sp-dataset/LJSpeech-1.1/metadata.csv"

wav_folder = "/kaggle/input/sp-dataset/LJSpeech-1.1/wavs"

df = pd.read_csv(csv_path, sep="|", header=None)

file_id = df.iloc[0, 0]   # first audio id

file_path = os.path.join(wav_folder, file_id + ".wav")

signal, sr = librosa.load(file_path, sr=None)

print("Loaded:", file_path)

plt.figure()

plt.plot(signal)

plt.title("Speech Waveform")

plt.xlabel("Samples")

plt.ylabel("Amplitude")

plt.show()slice_2sec = signal[:2*sr]

plt.figure()

plt.plot(slice_2sec)

plt.title("First 2 seconds (Sliced)")

plt.show()

normalized = slice_2sec / np.max(np.abs(slice_2sec))

plt.figure()

plt.plot(normalized)

plt.title("Normalized Signal")

plt.show()

amplified = signal * 2

deamplified = signal * 0.5

plt.figure()

plt.plot(amplified)
```

```
plt.title("Amplified Signal")
plt.show()
plt.figure()
plt.plot(deamplified)
plt.title("De-amplified Signal")
plt.show()
upsampled = resample(signal, len(signal)*2)
downsampled = resample(signal, len(signal)//2)
plt.figure()
plt.plot(upsampled)
plt.title("Upsampled Signal")
plt.show()
plt.figure()
plt.plot(downsampled)
plt.title("Downsampled Signal")
plt.show()
plt.figure()
plt.plot(signal, label="Speech")
# mark silence threshold
plt.axhline(y=threshold, color='r', linestyle='--')
plt.axhline(y=-threshold, color='r', linestyle='--')
plt.title("Voiced / Unvoiced / Silence (Visual)")
plt.legend()
plt.show()
```

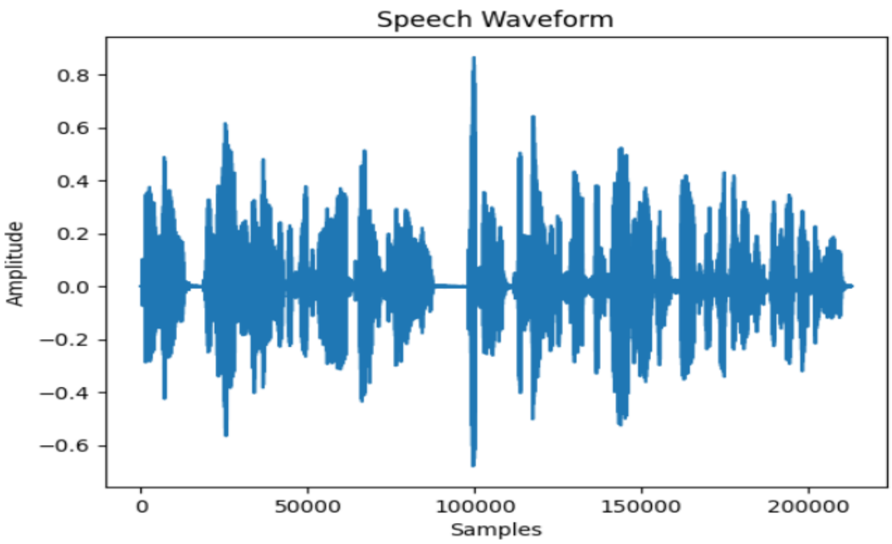
Output:

1. Audio playback of the recorded speech signal was successfully obtained.

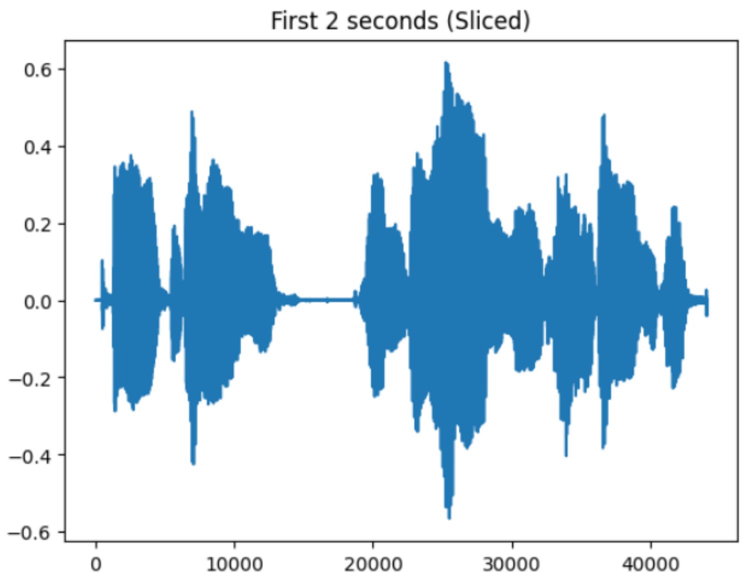
BL.EN.U4AIE21032
B.CHAITANYA KIRAN



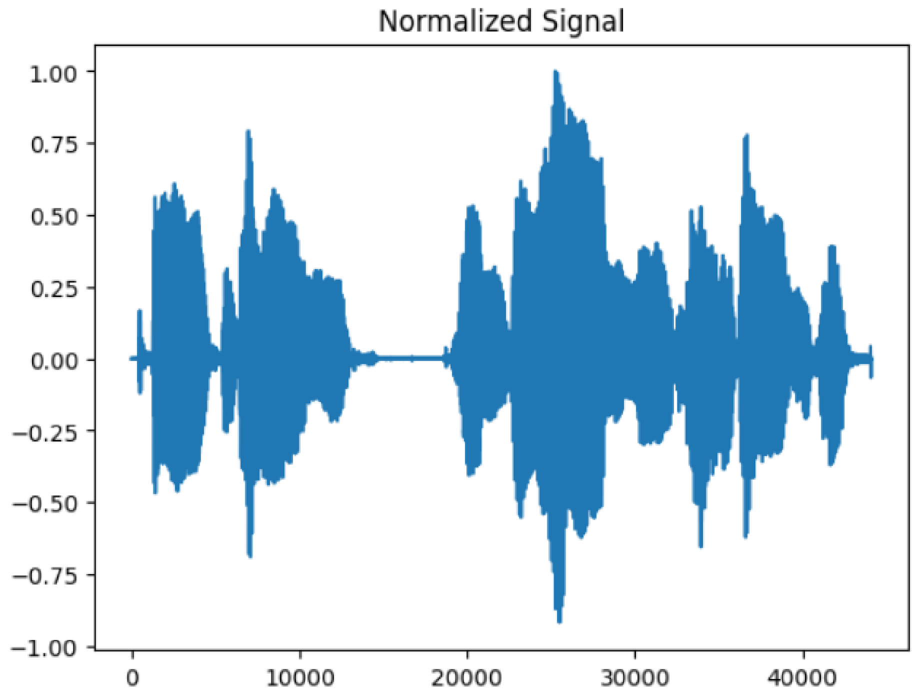
2. The time-domain waveform showed clear variations in amplitude corresponding to speech activity.



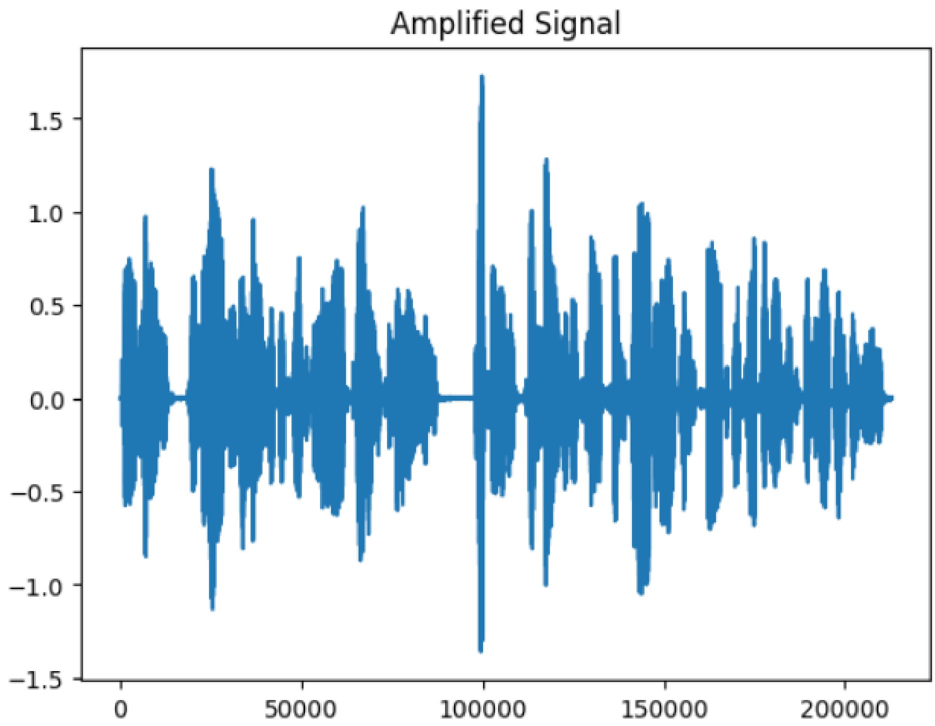
3. Sliced and normalized signals demonstrated amplitude scaling without distortion.



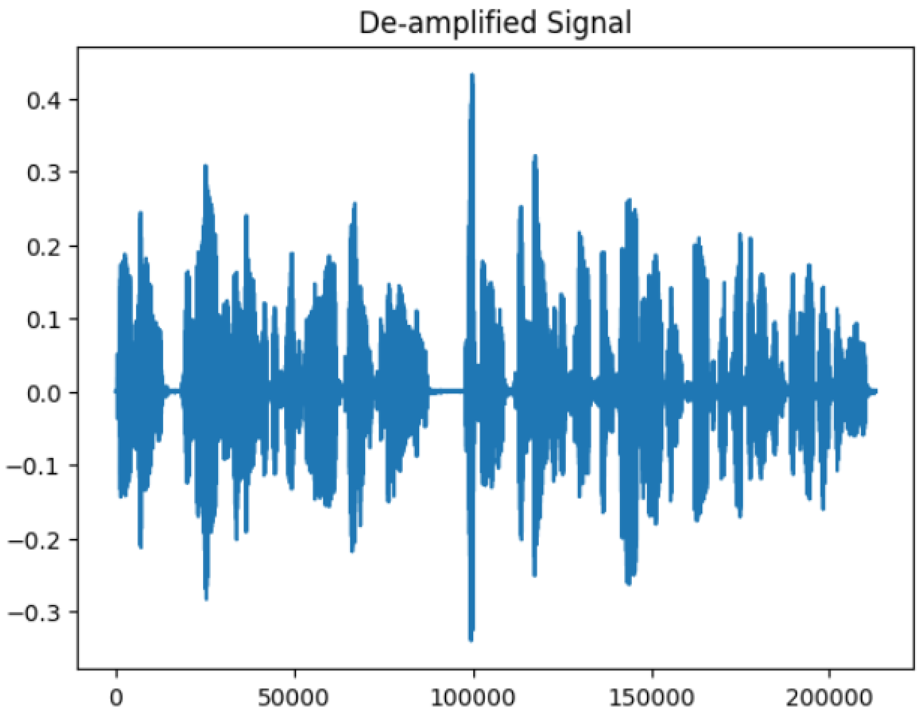
BL.EN.U4AIE21032
B.CHAITANYA KIRAN



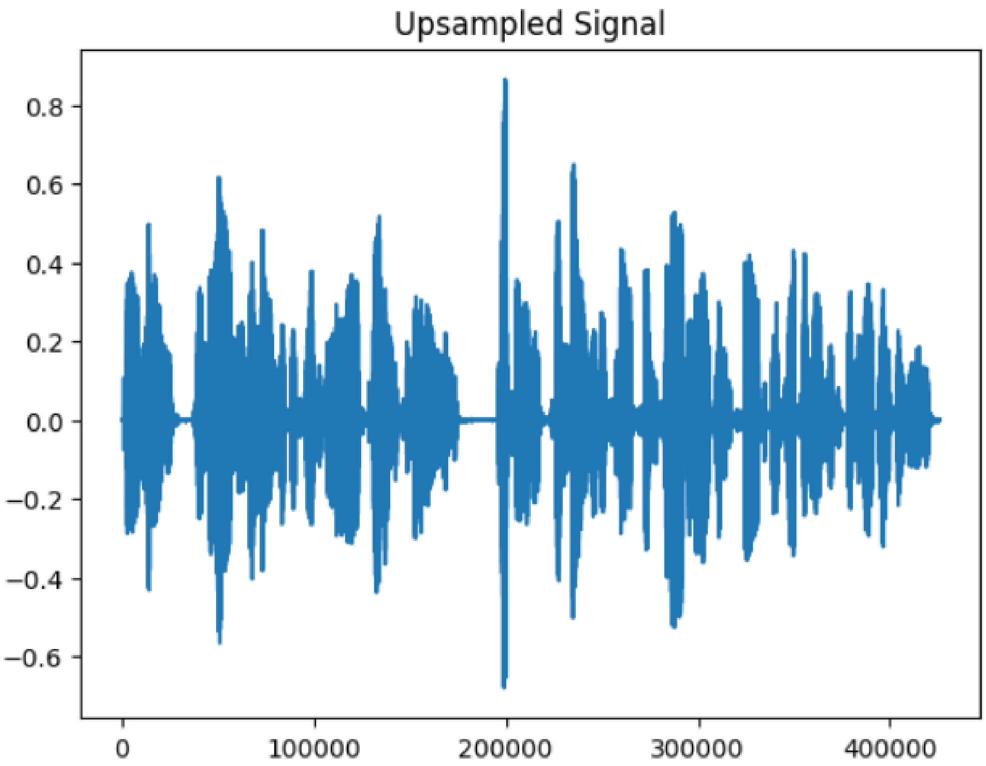
4. Amplification and attenuation showed proportional changes in signal strength.



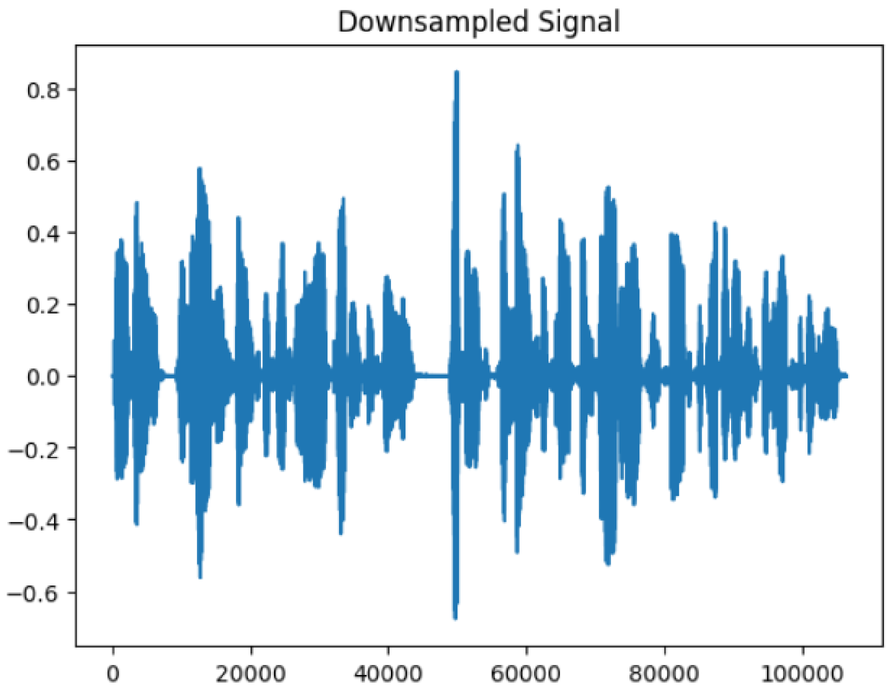
BL.EN.U4AIE21032
B.CHAITANYA KIRAN



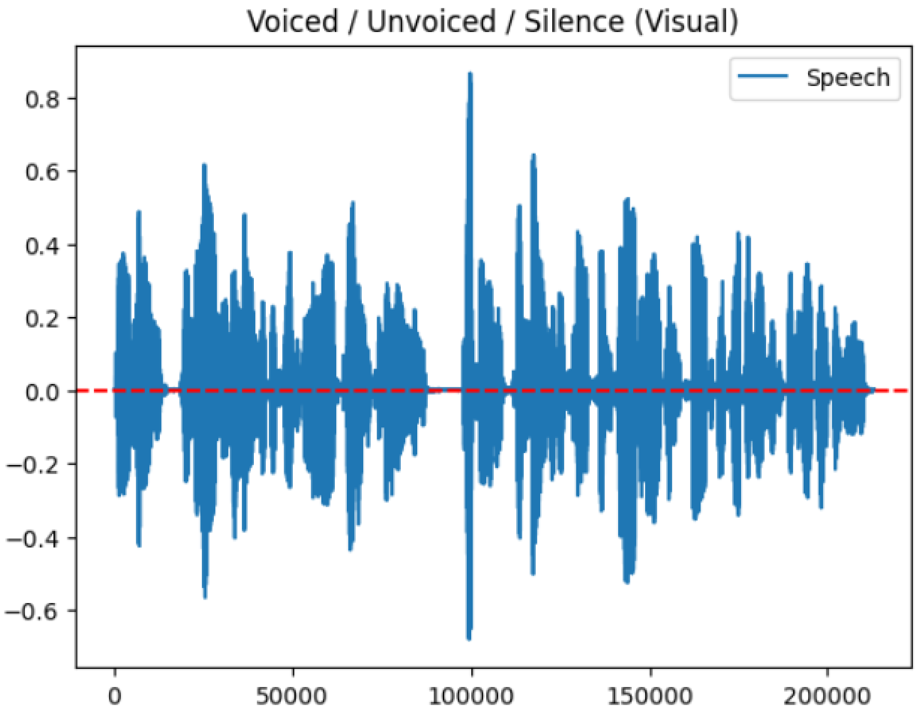
5. Up-sampling increased the number of samples, while down-sampling reduced temporal resolution.



BL.EN.U4AIE21032
B.CHAITANYA KIRAN



6. Voiced, unvoiced, and silence regions were visually distinguishable based on amplitude variations.



BL.EN.U4AIE21032
B.CHAITANYA KIRAN

Conclusion :

This experiment helped in understanding the basic structure of speech signals. The waveform visualization showed how amplitude varies over time. Operations like slicing, normalization, amplification, and resampling were successfully implemented. Voiced, unvoiced, and silence regions were visually distinguishable. Overall, the experiment improved practical understanding of fundamental speech processing techniques.

Result:

The objectives of the experiment were successfully achieved.