Current Project: Speech to text for a robot

Goals I'm aiming for:

- Speech to text for English
- Speech to text for Telugu
- Conversion of Languages
- Should not use out-sorced API(s)

Previous Methods:

- Using Librosa and Scipy (A)
- Using Speech Recognition
- Using Simple Transformers (B)

Current Methodology:

- Ditched the use of SpeechRecognition as it belongs to GoogleAPI
- Using Librosa and Scipy with enhancements
- Creating an output via combination of A and B

Data set I'm using:
https://www.kaggle.com/c/tensorflow-speech-recognition-challenge

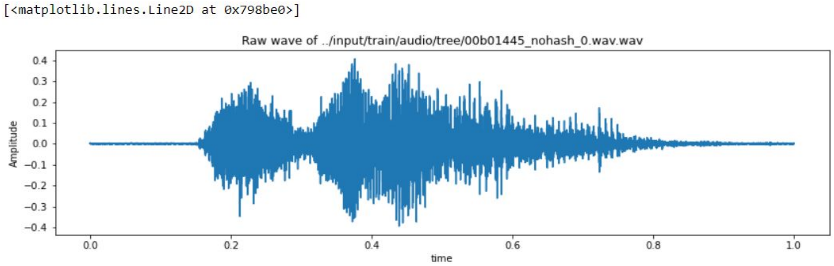
Libraries required:

Matplotlib, kiwisolver, cycler, Jit, librosa, IPython, scipy, numpy, numba (0.48.0 version only)
Pylint, Sklearn, keras, virtualenv, tensorflow, Pyaudio, SoundDevice, requests, SoundFile, darr

This consists of 2 parts: Part - **A1 : Importing and Data Visualization**

- Importing and Visualizing the audio using pyplot library (Source, integrity verification)

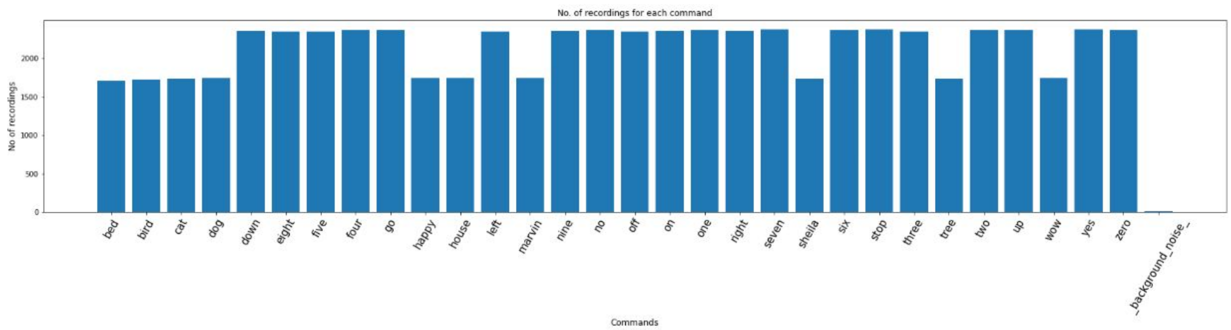
```
In [4]: train_audio_path = 'D:/Work/speech to text/tensorflow_English/train/audio/'
        samples, sample_rate = librosa.load(train_audio_path+'tree/00b01445_nohash_0.wav', sr = 16000)
        fig = plt.figure(figsize=(14, 8))
        ax1 = fig.add_subplot(211)
        ax1.set_title('Raw wave of ' + '../input/train/audio/tree/00b01445_nohash_0.wav.wav')
        ax1.set_xlabel('time')
        ax1.set_ylabel('Amplitude')
        ax1.plot(np.linspace(0, sample_rate/len(samples), sample_rate), samples)

Out[4]: [<matplotlib.lines.Line2D at 0x798be0>]
```



- Sampling the audio (checking the sampling rate)
- Resampling the audio

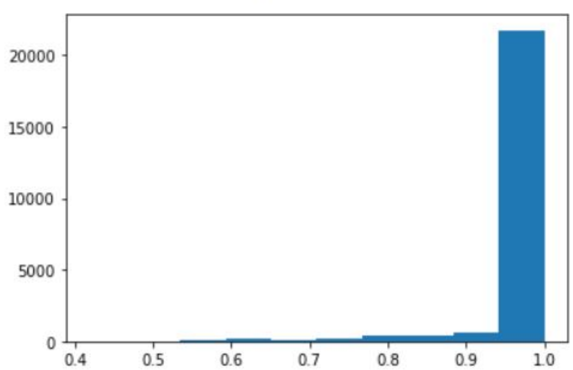
```
In [5]: ipd.Audio(samples, rate=sample_rate)
        print(sample_rate)

16000

In [6]: samples = librosa.resample(samples, sample_rate, 8000)
        ipd.Audio(samples, rate=8000)

Out[6]: 0:01 / 0:01
```

- Further establishing recordings of each command



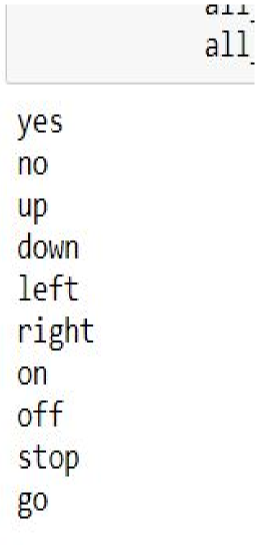
- Determining the duration of the audio recordings

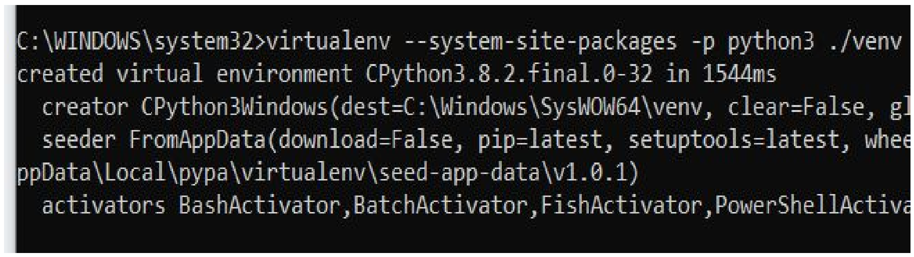
```
Out[9]: (array([1.5000e+01, 3.0000e+01, 4.4000e+01, 1.3800e+02, 1.3600e+02,
                1.7900e+02, 3.6600e+02, 4.3400e+02, 5.9300e+02, 2.1747e+04]),
         array([0.418 , 0.4762, 0.5344, 0.5926, 0.6508, 0.709 , 0.7672, 0.8254,
                0.8836, 0.9418, 1.    ]),
         <a list of 10 Patch objects>)
```



A2: Processing the Audio

- Establishing and trimming commands to match time requirements
- Converting outputs to encoded integers
- Converting encoded labels to vectors
- Generating 3D input set for use in Model
- Establishing training and validation model boundaries

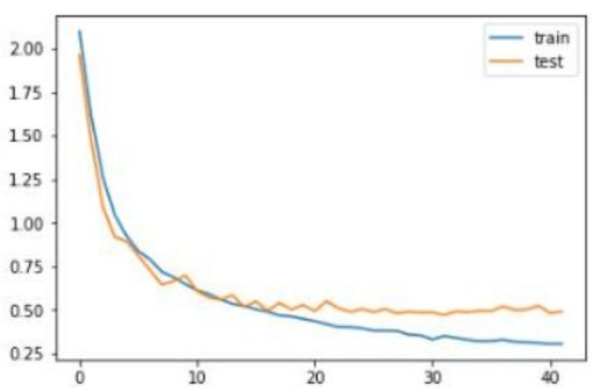




A3: Creating a Model Architecture

Currently using a CNN based Conv1d

- Creating model using keras functional API
  My model consists of
    - 3 Conv1d Layers
    - Flat Layer
    - 2 Dense Layers
- Establishing a loss function
- Establishing Model checkpoints
- Training model on batch size of 16
- Generating a diagnostic plot



- Extrapolating the best model

A3: Adding your own audio

Further libraries: Wave, Sys,

- Include microphone as a source in PyAudio
- Convert audio into .wav format
- Read dataframes
- Further Input methods here: https://people.csail.mit.edu/hubert/pyaudio/docs/
- Testing the audio

```
Recording Audio....
Writing Audio...
1B,
1D,
1C,
I predict user said 'Left'
```

Note:

- Currently using SoundDevice and SoundFile instead of PyAudio
- PyAudio is NOT being used (Refer to version history for all updates on this)

VERSION HISTORY : LIST OF CHANGES MADE AFTER THE ABOVE REPORT

0.1 - Testing and establishing more data sets
0.2 - Using Librosa library in more places
0.21 - Used multi-classification approach
0.3 - Improving sampling rate, durational requirements to increase speed via scipy
0.31 - Forcedlibrary versions to prevent compatibility errors
0.4 - Added support to use Tensorflow backend libraries form system sites (--systemsite)
0.41- Created and activated new virtual environment
0.5 - Added a fourth Conv1d Layer
0.6 - Enhanced loss function
0.61 - Implemented EarlyCheckpoints (MANDATORY)
0.62 - Increased Training Model batch size to 32 (Efficient and faster)
0.7 - Using SoundFile, SoundDevice instead of PyAudio
0.71 - Function to record and writing the data
0.72 - Established sample rate as 16000