* **MELD dataset**
  + Before this, there was not any multi-party emotional conversation database
  + Emotion(7) + sentiment(3) annotation
  + Audio, visual and textual modalities
  + Parallel data channels
  + Sequence and context modeling is important
  + Being multimodal improves context modeling and hence overall emotion recognition
* **THE SPEECH EMOTION PROBLEM TASK**
* The first thing that I noticed about the data was there is distinctive background noise and specifically the laugh. Also, the length of all the audio recordings varies a lot.
* Understanding WAV file format

<https://blogs.msdn.microsoft.com/dawate/2009/06/23/intro-to-audio-programming-part-2-demystifying-the-wav-format/>

-Chunks are used to represent certain metadata about the file

-Channels in .wav file format

-The data provided is 2 channel data that is it is stereo data

-Bit Depth: if 16 then amplitude can take 2^16 diff values

* Step 1: Visualizing the speech data

<https://www.youtube.com/watch?v=Z7YM-HAz-IY>

DSP background study for preprocessing the audio data

1. FFT Fast Fourier Transform

<https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft>

Gives us a periodogram

Basically, we shift from the time domain to frequency domain when we take FFT of the signal.

1. Spectrogram

Taking Periodograms and stacking them adjacent to each other in time. The intensity of the pixel value is a measure of the amount that specific frequency contributes.

1. Short-Time Fourier Transform (STFT)

Take a specific period in time (window length) and perform FFT over that window in time.

1. MEL FILTERBANK

It is based on the concept that humans can differentiate between frequencies better when they are low as compared to when they are high.

Used to build features based on power spectral density

However, the features extracted by this method are highly correlated. Discrete cosine transform is performed on the filterbank energy to get decorrelated coefficients.

<https://www.kaggle.com/davids1992/speech-representation-and-data-exploration>

Basically, this is a step towards exploring the data. I just googled visualizing speech data in python and this link popped up. Since I have not worked with speech data before it seemed quite informative to me. Even though it is aimed at speech recognition and not emotion recognition I tried to understand and implement all the techniques mentioned here on the MELD audio data.

1. Plotting a Spectrogram using Scipy library:

A spectrogram is a visual representation of the spectrum of frequencies as it varies with time. DFT (Discrete Fourier transform) is used to compute the same. There exist various types of windowing functions in DFT, in particular, the Hanning window is used here.

1. <http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/>

MFFC the Kaggle post mentions it so I searched about it. However, after searching about I found that it is useful for extracting features for speech recognition only and it removes emotional features so most probably it will not be useful in our case. I still try that as well as a found Research papers using MFCC for emotion recognition

* Some SER Projects on Github used for reference

<https://github.com/x4nth055/emotion-recognition-using-speech>

<https://github.com/amanbasu/speech-emotion-recognition>

* Step2: Removing Noise from the data

<https://timsainburg.com/noise-reduction-python.html>

Amazing tutorial on noise reduction. However, they are adding noise and removing it which is not making sense to me. Also

* Step3: Feature Selection

<https://www.intechopen.com/online-first/automatic-speech-emotion-recognition-using-machine-learning>

Read it and found out that other features can be used as well like MSF after that we can also choose relevant features from MFCC and MSF. I will try this if time is left in the end

<https://www.microsoft.com/en-us/research/publication/high-level-feature-representation-using-recurrent-neural-network-for-speech-emotion-recognition/>

Gave read

<https://www.thepythoncode.com/article/building-a-speech-emotion-recognizer-using-sklearn>

Implemented and test the features used here.

* Step4: Training an MLP classifier

Trained a basic MLP classifier by taking the mean and then stacking MFCC coefficients and also ignoring the audio clips that were less than 1s in length. As a result, I got only 50% accuracy which is pretty bad.

My opinion on this is

1. Instead of using MLP use of CNN will improve the accuracy as MFFC seemed to sort of images to be and CNN
2. Reduce the laughter noise
3. Experiment with other possible features
4. I feel that the words that they are saying mostly is not really indicating their feeling in happy(not sure )

* Step5: Trying to remove laughter background

1. Cropped out the laughter from a specific training example
2. <https://timsainburg.com/noise-reduction-python.html>

Now use this method to remove the laughter not sure though if it will work or not

<https://github.com/timsainb/noisereduce>

Not working the way I want it to

* Step6: Training a CNN

Creating a sequential convolutional model in Keras.

And train it however since the model trained is not complex and lack of training examples the model seems to be overfitting as it is giving high accuracy of the training data but the accuracy fluctuates on the validation data