**Emotion Detection through Speech**

**Introduction:**

Detecting emotions is one of the most important marketing strategy in today’s world. You could personalize different things for an individual specifically to suit their interest. For this reason, we decided to do a project where we could detect a person’s emotions just by their voice which will let us manage many AI related applications. Some examples could be including call centers to play music when one is angry on the call. Another could be a smart car slowing down when one is angry or fearful. As a result, this type of application has much potential in the world that would benefit companies and even safety to consumers.

**Data Used:** We got audio datasets with around 2000 audio files which were in the wav format from the following websites: <http://neuron.arts.ryerson.ca/ravdess/?f=3>, <http://kahlan.eps.surrey.ac.uk/savee/Download.html>

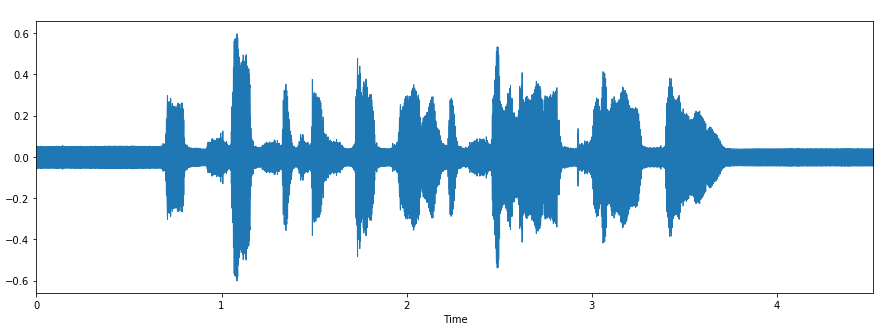
The first website contains speech data which is available in three different formats.

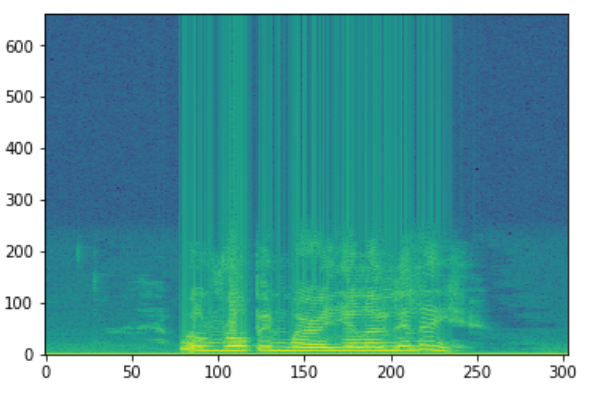
1. Audio Visual – Video with speech
2. Speech – Audio only
3. Visual – Video only

We went with the Audio only zip file because we are dealing with finding emotions from speech. The zip file consisted of around 1500 audio files which were in wav format.

The second website contains around 500 audio speeches from four different actors with different emotions.

We tested out one of the audio files to know its features by plotting its waveform and spectrogram.





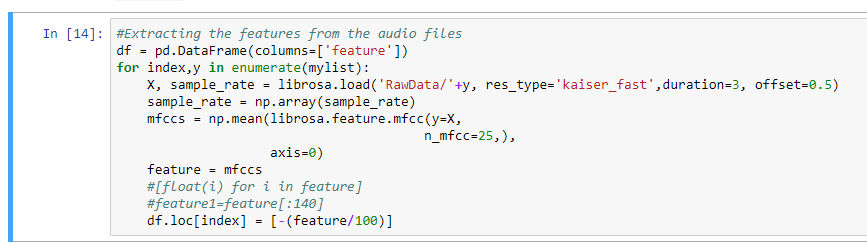
The next step involves organizing the audio files. Each audio file has a unique identifier at the 6th position of the file name which can be used to determine the emotion the audio file consists. We have 5 different emotions in our dataset.

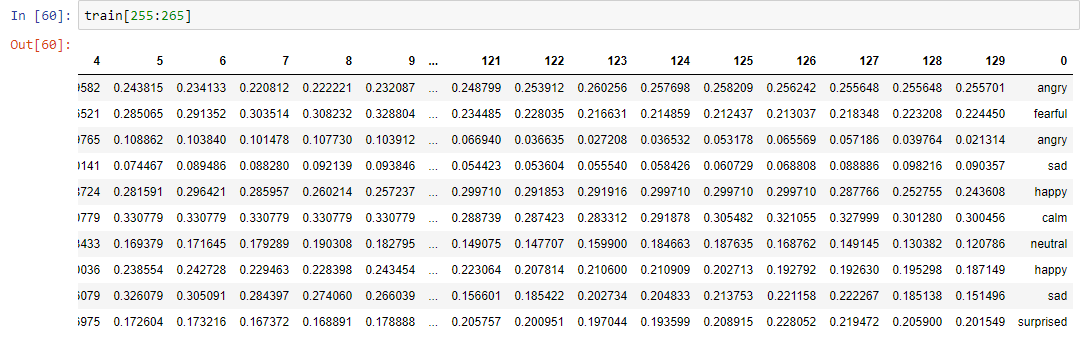
1. Calm
2. Happy
3. Sad
4. Angry
5. Fearful

We used Librosa library in Python to process and extract features from the audio files. Librosa is a python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems. Using the librosa library we were able to extract features i.e MFCC(Mel Frequency Cepstral Coefficient). MFCCs are a feature widely used in automatic speech and speaker recognition. We also separated out the females and males voice by the using the identifiers provided in the website. This was because as experiment we found out that separating male and female voices increased by 15%. It could be because of the pitch of the voice was affecting the results.

Each audio file gave us many features which were basically array of many values. These features were then appended by the labels which we created in the previous step.

The next step involved dealing with the missing features for some audio files which were shorter in length. We increased the sampling rate by twice to get the unique features of each emotional speech. We didn’t increase the sampling frequency even more since it might collect noise thus affecting the results





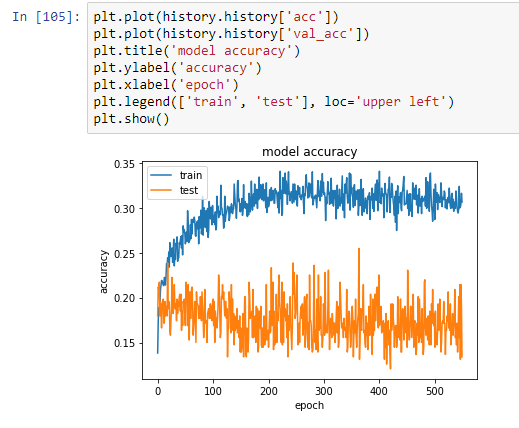
The next steps involve shuffling the data, splitting into train and test and then building a model to train our data.

We built a Multi Perceptron model, LSTM model and CNN models. The MLP and LSTM were not suitable as it gave us low accuracy. As our project is a classification problem where were categorize the different emotions, CNN worked best for us.

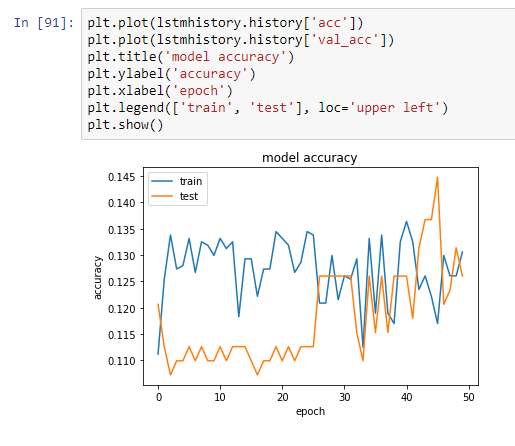
**Analysis:**

1. **MLP Model:** The MLP model we created had a very low validation accuracy of around 25% with 8 layers, softmax function at the output, batch size of 32 and 550 epochs.

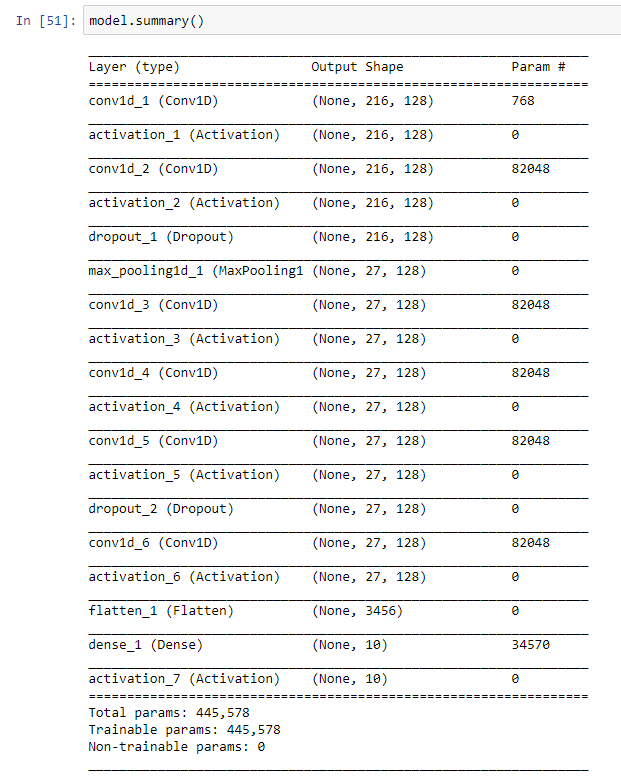


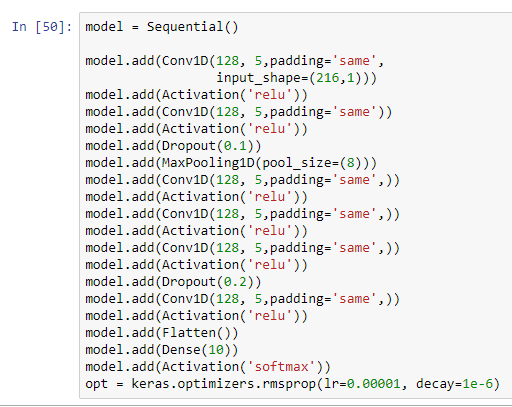


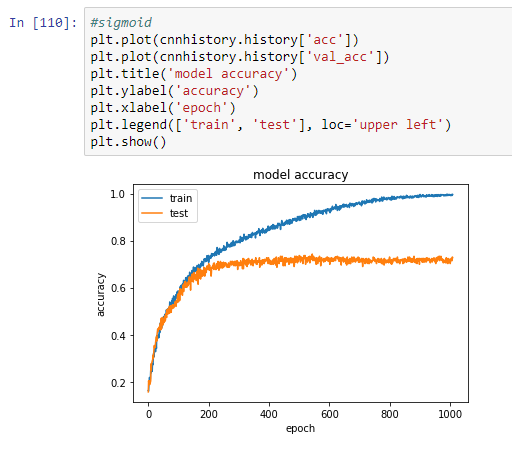
1. **LSTM:** The LSTM model had the lowest training accuracy of around 15% with 5 layers, tan h activation function, batch size of 32 and 50 epochs



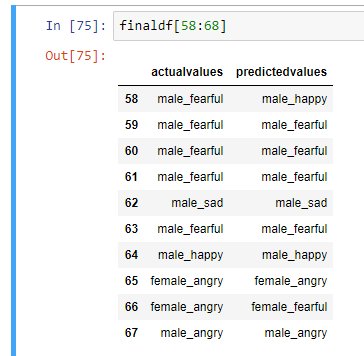
1. **CNN :** CNN model was the best for our classification problem. After training numerous models we got the best validation accuracy of 60% with 18 layers, softmax activation function, rmsprop activation function, batch size of 32 and 1000 epochs







After training the model we had to predict the emotions on our test data. The following picture shows our prediction with the actual values.



**Conclusion:** After building numerous different models, we have found our best CNN model for our emotion classification problem. We achieved a validation accuracy of 70% with our existing model. Our model could perform better if we have more data to work on. What’s more surprised is that the model performed excellent when distinguishing between a males and females voice. We can also see above how the model predicted against the actual values. In the future we could build a sequence to sequence model to generate voice based on different emotions. E.g. A happy voice, A surprised one etc.