Speech Emotion Recognition

Ayush Anand (B20CS082)* and Vikash Yadav (B20AI061)*
*IIT Jodhpur CSE, email: anand.5@iitj.ac.in
*IIT Jodhpur AI, email: yadav.41@iitj.ac.in

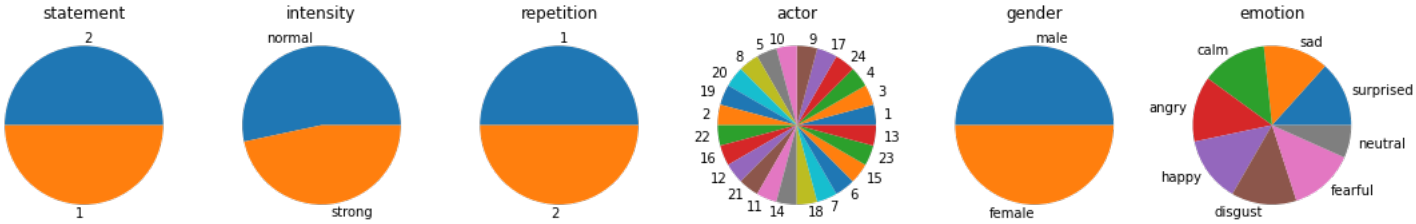
Abstract

Speech emotion recognition aims to identify the emotion expressed in the form of speech in an audio clip. Use of Machine Learning in the area of emotion recognition is a relatively nascent research area. This document aims to provide an approach towards speech emotion recognition through machine learning models. The dataset chosen for the same is the RAVDESS[1].

I. Methodology

A. Dataset Procurement and Analysis

The dataset used for this project is the RAVDESS[1]. It contains 1440 audio clips, with 24 actors each vocalizing two lexically-matched statements. There are 60 trials per actor. 12 of the actors are male and 12 of them are females.



The dataset description can be found by clicking here.

B. Feature Engineering

AI models only understand numbers. So after having the audifiles ready, we must first represent them as numbers. We do so by loading them as an array that describes the change in amplitude over time. Now these numbers take us no where as there is no direct way to classify the emotion based off the values of this array. So we engineer it to get some other acoustic features that are of use to us. We apply Fourier Transforms to go from time to frequency domain and obtain **spectograms** which are information of frequency, and amplitude along time.

We consider few different spectograms as our features for the audio:

- MFCC (Mel Frequency Cepstral Coefficients)
- chroma
- melspectogram

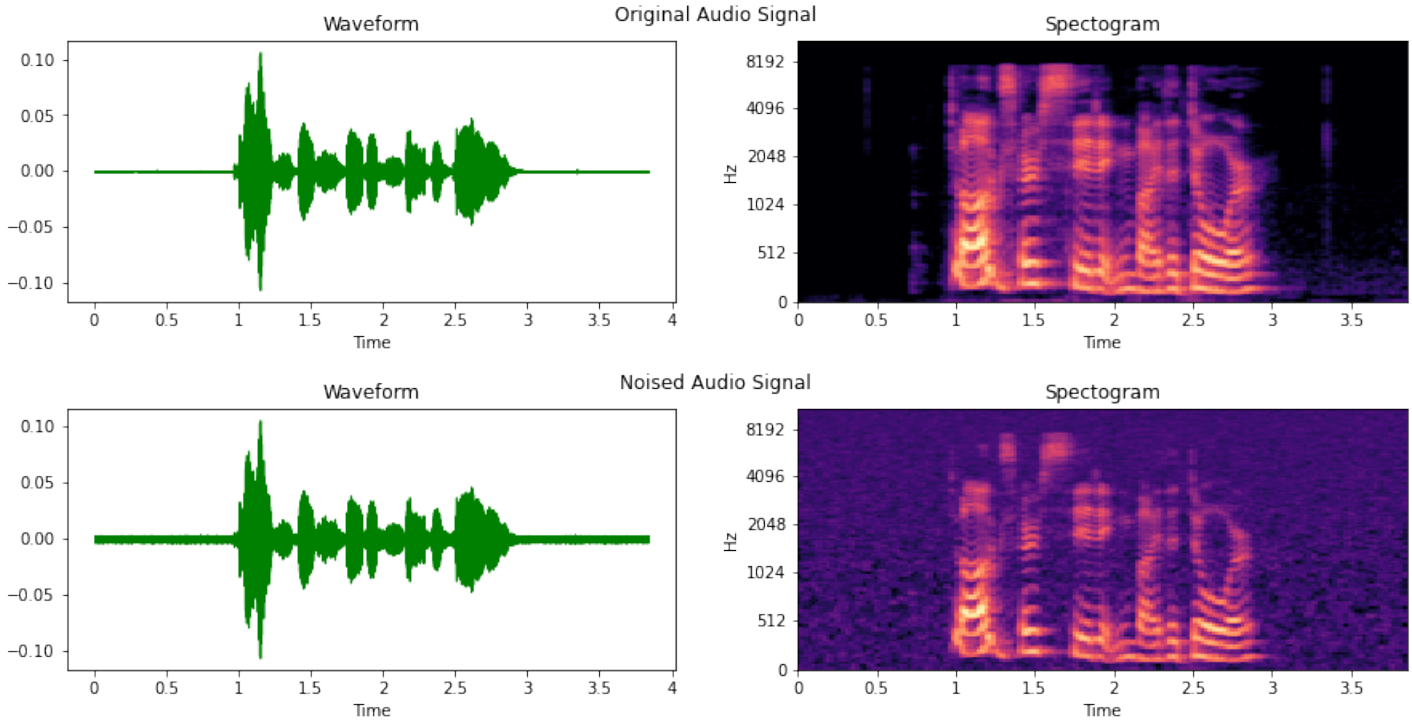
1) MFCC: The mel scale approximates the human auditory systems response more efficiently in comparision to other scales. MFCC's are frequently used in the context of feature extraction from audio clips. MFCC is obtained from the cepstrum of the signal with frequencies scaled based on mel scale.

2) Chroma: Chroma is similar to MFCC except that we use a 12-tone scale based on the 12 pitches used in tonal music, further we do not use cepstral unlike MFCC.

3) Melspectogram: Similar to MFCC but the process is applied on the spectrum and not on the cepstral of the signal.

C. Preprocessing

MFCC values are not very robust in the presence of additive noise and hence we normalize their values while using them. Another reason to normalize the extracted features is to ensure that all features are treated with equal importance when being passed on to a model.
We also added some noise to our audio signals to see if that gave any better results.



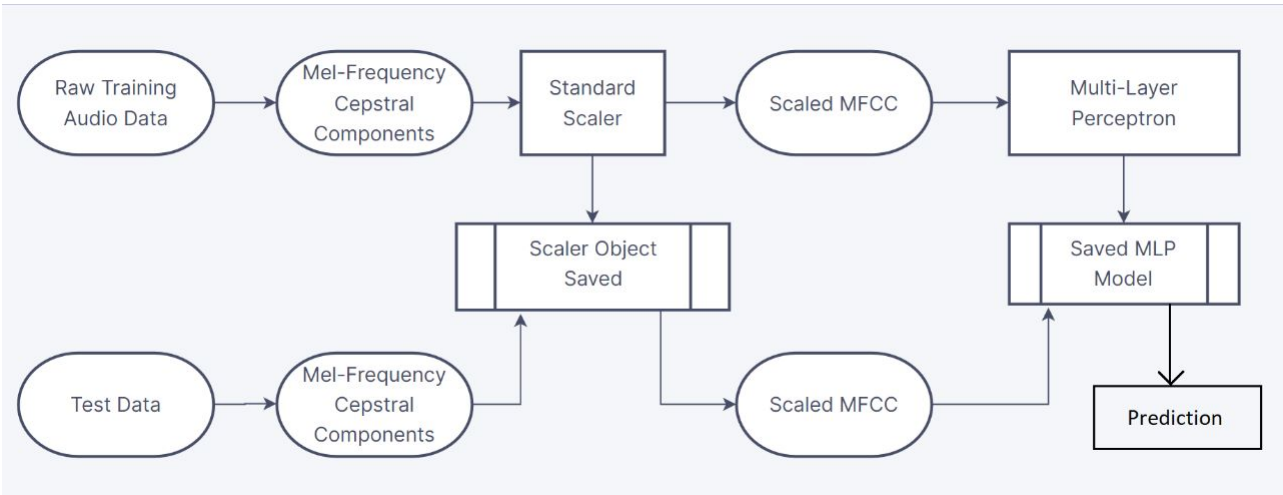
D. Models

We tested several models on several combination of features. We made different datasets for the 3 spectorgram features and one with all 3 combined and both with and without noise, so in total 8.
We applied the follwing models:

- MLP(MultiLayerPerceptron) (activation:reLU, hidden_layer_sizes:(320,160,80,40))
- Support Vector Machines (C:1000)
- Logistic Regression (solver:lbfgs, max_iter:1000)
- XGBoost (learning_rate:0.3, max_depth:10)
- LightBGM (learning_rate:0.3, num_leaves:20,n_estimators:1000, min_child_samples:15)
- K-Nearest Neighbours (n_neighbors:5)

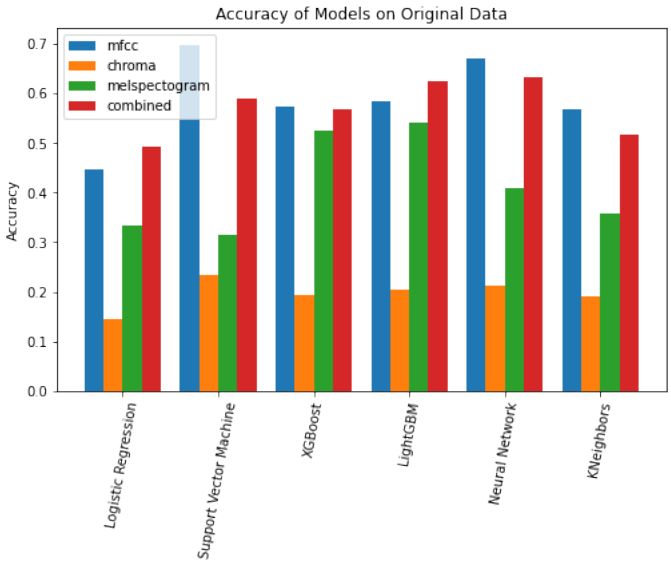
Then we tuned the parameters of the 3 best performing models and then chose the best one for making the final pipeline and deployment.
All the results of performance are attached below.

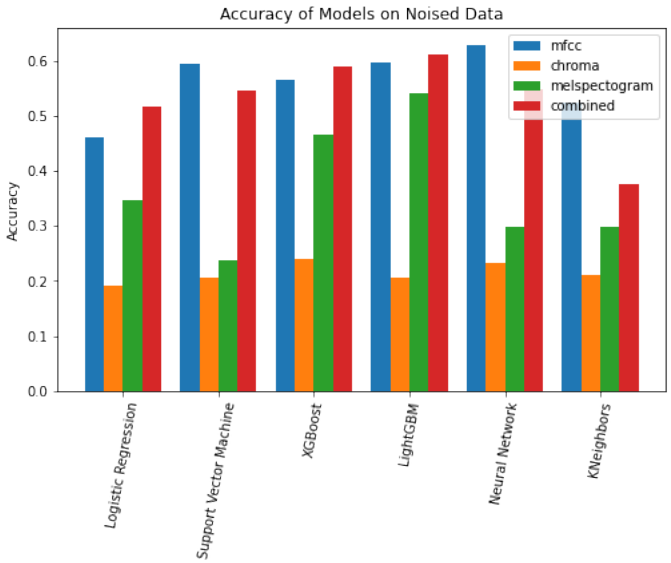
E. Final Pipeline



II. Results and Analysis

A. Accuracy





MODEL	MFCC		CHROMA		MEL SPEC		COMBINED	
	Original	Noised	Original	Noised	Original	Noised	Original	Noised
Logistic Regression	0.4479	0.4618	0.1458	0.1910	0.3333	0.3472	0.4931	0.5174
Support Vector Machine	0.6979	0.5938	0.2326	0.2049	0.3160	0.2361	0.5903	0.5451
XGBoost	0.5729	0.5660	0.1944	0.2396	0.5243	0.4653	0.5694	0.5903
LightGBM	0.5833	0.5972	0.2049	0.2049	0.5417	0.5417	0.6250	0.6111
Neural Network	0.6701	0.6285	0.2118	0.2326	0.4097	0.2986	0.6319	0.5486
KNeighbors	0.5694	0.5243	0.1910	0.2118	0.3576	0.2986	0.5174	0.3750

B. *Parameter Tuning for best 3 models*

```
Best Accuracy: 0.6979166666666666
Best Parameters: {'C': 2000, 'kernel': 'rbf'}
```

(a) Best Params

	precision	recall	f1-score
0	0.62	0.68	0.65
1	0.74	0.92	0.82
2	0.71	0.63	0.67
3	0.82	0.69	0.75
4	0.77	0.74	0.76
5	0.59	0.65	0.62
6	0.59	0.59	0.59
7	0.67	0.65	0.66
accuracy			0.70
macro avg	0.69	0.69	0.69
weighted avg	0.70	0.70	0.70

(b) Scores

Fig. 1: Tuning SVM

```
Best Accuracy: 0.7118055555555556
Best Parameters: {'activation': 'tanh', 'hidden_layer_sizes': (320, 160, 80, 40)}
```

(a) Best Params

	precision	recall	f1-score
0	0.71	0.63	0.67
1	0.80	0.84	0.82
2	0.71	0.58	0.64
3	0.74	0.74	0.74
4	0.83	0.80	0.82
5	0.55	0.65	0.59
6	0.62	0.68	0.65
7	0.69	0.71	0.70
accuracy			0.71
macro avg	0.71	0.70	0.70
weighted avg	0.72	0.71	0.71

(b) Scores

Fig. 2: Tuning MLP

```
Best Accuracy: 0.5729166666666666
Best Parameters: {'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 500}
```

(a) Best Params

	precision	recall	f1-score
0	0.54	0.37	0.44
1	0.57	0.74	0.64
2	0.61	0.47	0.53
3	0.57	0.67	0.61
4	0.77	0.68	0.72
5	0.53	0.59	0.56
6	0.45	0.41	0.43
7	0.47	0.52	0.49
accuracy			0.57
macro avg	0.56	0.55	0.55
weighted avg	0.58	0.57	0.57

(b) Scores

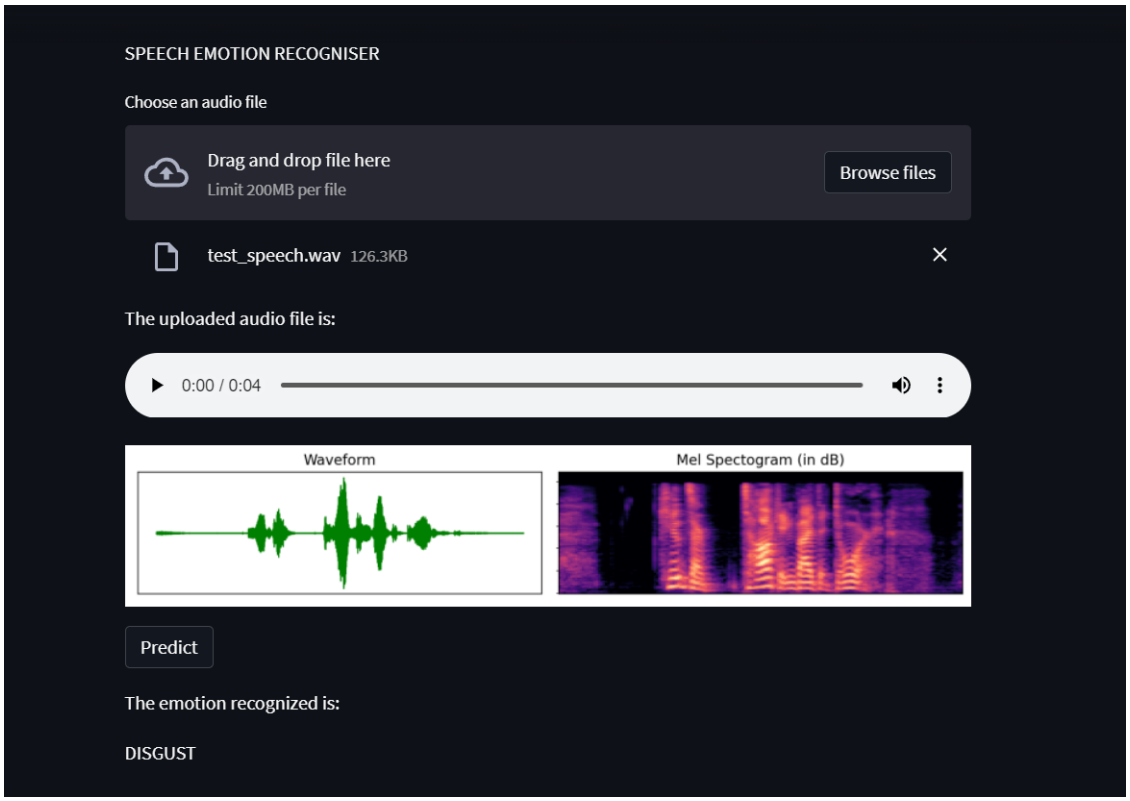
Fig. 3: Tuning XGBoost

III. Contributions

Vikash Yadav (B20AI061) : Pre-Processing, Feature Engineering, Trying & Testing Models, Report.
Ayush Anand (B20CS082) : Feature Exploration, Parameter Tuning, Web Deployment, Report.

IV. Appendix (Deployment)

We deployed our final model that we stored using pickle, in the form of a website that let's you upload a audio file and will predict the emotion for you. It also visualises the audio file in form of wave and spectogram.



The Website: https://share.streamlit.io/iamayushanand/ser/main/streamlit/app.py
Code for the same: https://github.com/iamayushanand/SER

REFERENCES

[1] Steven R. Livingstone and Frank A. Russo. Ravdess emotional speech audio, 2019.