Speech Emotion Recognition

Dhruv Mahajan(B20EE016), Haardik Ravat (B20EE021)
Github Repo: PRML-project

Abstract: *This paper reports our experience with building a speech-emotion recognition model using machine learning. Using the RAVDESS dataset, multiple classifications and deep learning models are trained on the data and compared. Furthermore, techniques like data augmentation were used to get better models and predictions. The best model was also deployed on the web for an interactive user interface.*

INTRODUCTION

Speech is the most common way humans have expressed and used to express their emotions. In today's post covid era when we had to rely on other digital communication forms like email, text there is a significant lack of an effective way to fully understand emotions associated with these messages hence detection and analysis of the same are vital importance in today's digital world of remote communication. In this project, we have used various machine and deep learning concepts taught in the course of PRML CSL2050 to detect underlying emotions in a recorded speech by analyzing the features of the audio recordings. Such a system can find use in a wide variety of application areas like analyzing caller-agent conversations etc.

DATA DESCRIPTION AND PREPROCESSING

Finding emotions from the speech of the speakers can be done using 3 major features namely the visual features (the expressions of the speaker), lexical features (vocabulary used by the speaker), and acoustic features (sound properties like pitch, tone.) In the following project we are working on the RAVDESS dataset i.e we are using acoustic features to determine the emotions

Dataset Description :

This portion of the RAVDESS contains 1440 files: 60 trials per actor x 24 actors = 1440. The RAVDESS contains 24 professional actors (12 female, 12 male), vocalizing two lexically-matched statements in a neutral North American accent. Speech emotions include calm, happy, sad, angry, fearful, surprised, and disgusted expressions. Each expression is produced at two levels of emotional intensity (normal, strong), with an additional neutral expression. Each of the 1440 files has a unique filename. The filename consists of a 7-part numerical identifier (e.g., 03-01-06-01-02-01-12.wav). These identifiers define the stimulus characteristics:

Filename identifiers:

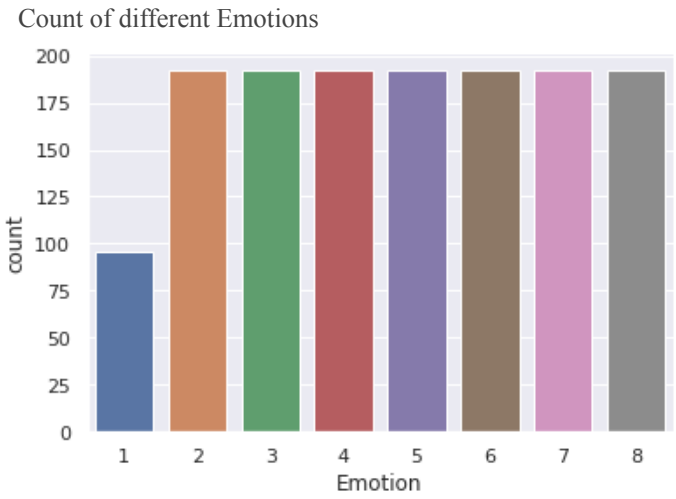
- Modality (01 = full-AV, 02 = video-only, 03 = audio-only).
- Vocal channel (01 = speech, 02 = song).
- Emotion (01 = neutral, 02 = calm, 03 = happy, 04 = sad, 05 = angry, 06 = fearful, 07 = disgust, 08 = surprised).
- Emotional intensity (01 = normal, 02 = strong). NOTE: There is no strong intensity for the 'neutral' emotion.
- Statement (01 = "Kids are talking by the door", 02 = "Dogs are sitting by the door").
- Repetition (01 = 1st repetition, 02 = 2nd repetition).
- Actor (01 to 24. Odd-numbered actors are male, even-numbered actors are female).

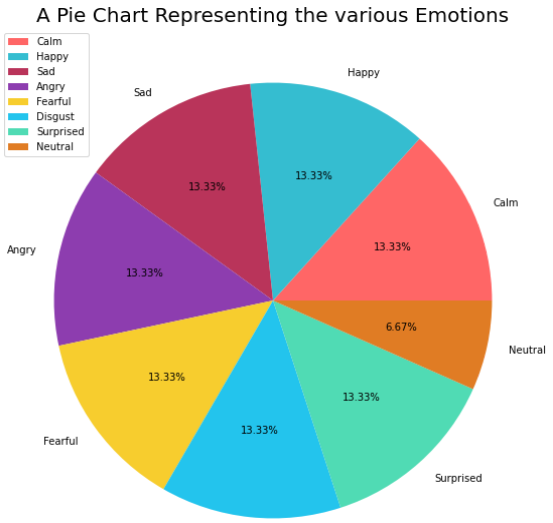
PREPROCESSING

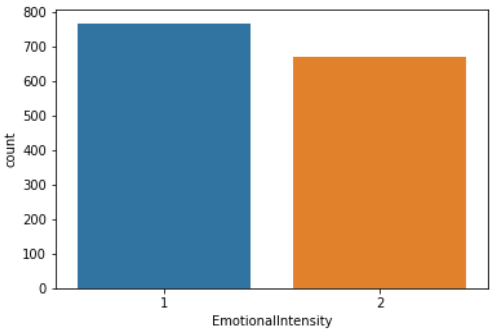
The dataset folder was iterated using the name of the .wav of the recordings. The training data frame was created with the paths of the audio recordings and classification information including Actor, Modality, VocalChannel, Emotional Intensity, Statement, Repetition, Emotion, and Gender as per the dataset information.

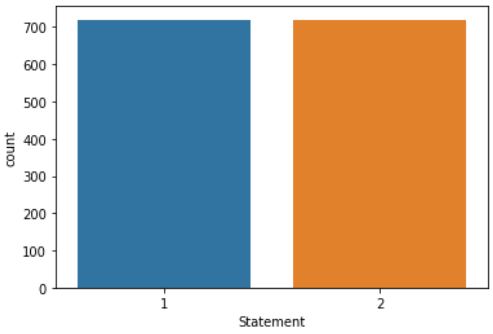
The aforementioned dataset, hence formed, was then iterated upon to get the following plots.

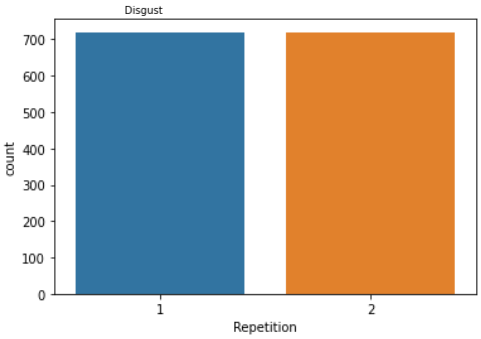
DATA VISUALIZATION





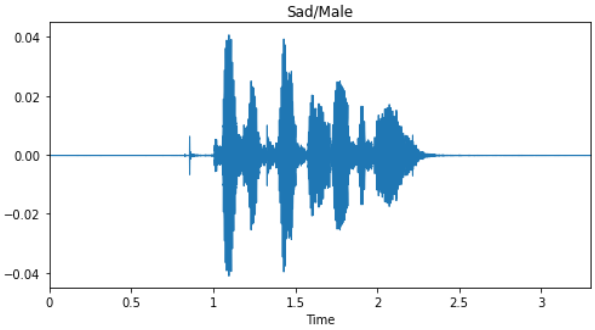


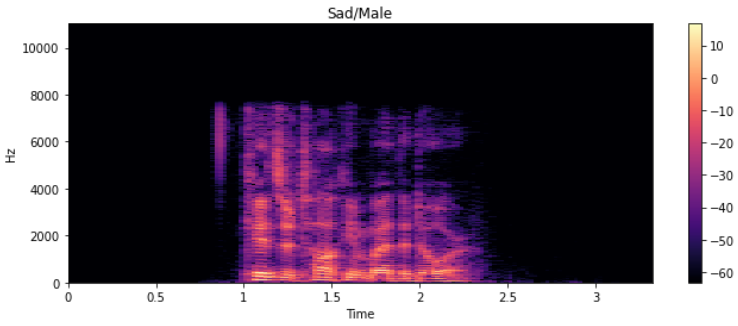




FEATURE EXTRACTION

The dataset formed previously was used as a source of paths for each audio file and was passed through a function named ‘extract_mfcc’ which uses the librosa library to get the mfcc for each audio file and append them in a list to form our classifying data frame. Using n_mfcc as 128, a panda data frame was created using the output data with 128 columns harboring the values of mfcc. Finally, another column named ‘Emotions’ was added to the data frame which was the predicted labels for the corresponding mfcc values using the initial path.
The output file and corresponding spectrogram is shown below:





MACHINE LEARNING MODELS

- **LightGBM**
  Tree-based learning algorithm that uses gradient boosting framework that uses tree-based learning algorithms. Extracted feature dataset file was trained and the recorded accuracy was **59%**
  To improve accuracy parameter tuning was done for hyperparameters (max-depth, learning rate, n-estimators, subsample, etc)
  Final Accuracy **62%**

- **Random Forest Classifier**
  Random Forest Classifiers use boosting ensemble methods to train upon various decision trees and produce aggregated results. The initial Accuracy reported was **52%.** With hyperparameter tuning, the final accuracy was **54%.**

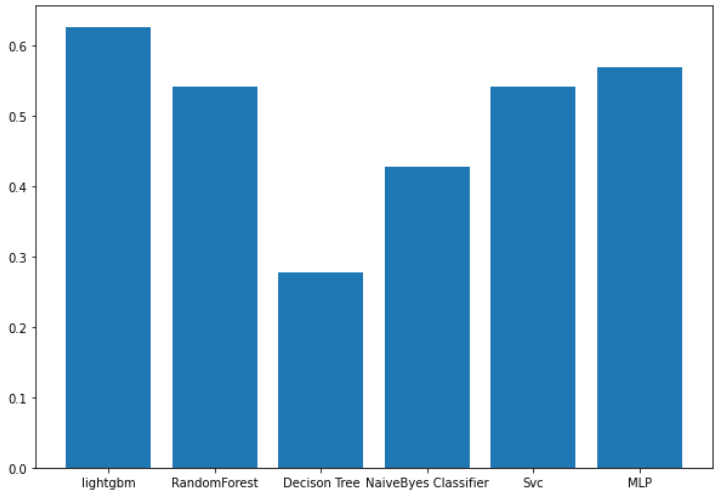
- **Decision Tree**
  A decision tree algorithm is a supervised machine learning technique. This algorithm can be used for classification as well as regression.
  Initial Accuracy **27%.** With hyper-parameters tuning final accuracy reported was **28%**

- **Naive Bayes Classifier**
  It is a probabilistic machine learning model that's used for the classification task. The main principle; behind this is the Bayes theorem.
  Accuracy reported: **43%**

- **Support Vector Classifier**
  SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes. This best decision boundary is called a hyperplane.
  Initial Accuracy: **42%**
  With hyperparameter tuning, final Accuracy was reported: **54%**

- **Multi-layer Perceptron**
  MLPClassifier stands for Multi-layer Perceptron classifier. Unlike other classification algorithms, it relies on an underlying Neural Network to perform the task of classification
  Accuracy reported:**57%**

A comparative analysis between models can be seen here:

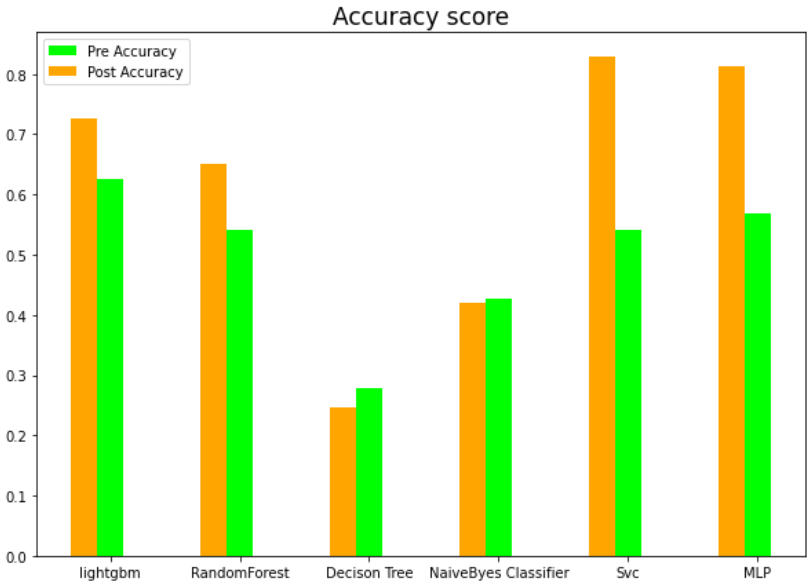


DATA AUGMENTATION

Data augmentation is a strategy that enables practitioners to significantly increase the diversity of data available for training models, without actually collecting new data. Here using the sound files for each path provided in the dataset, the noise was added to the sounds. The sounds thus formed were followed by calculating the mfcc for every audio file and added to the dataset. This increased the variation of dataset used.

For doing the same, the audio was converted into a numpy array and the noise factor was multiplied by each element. The thus formed numpy array was again converted into sound and used to calculate mfcc to add to the dataset.

Here is a comparative analysis of the approaches of data taken(with and without augmentation):



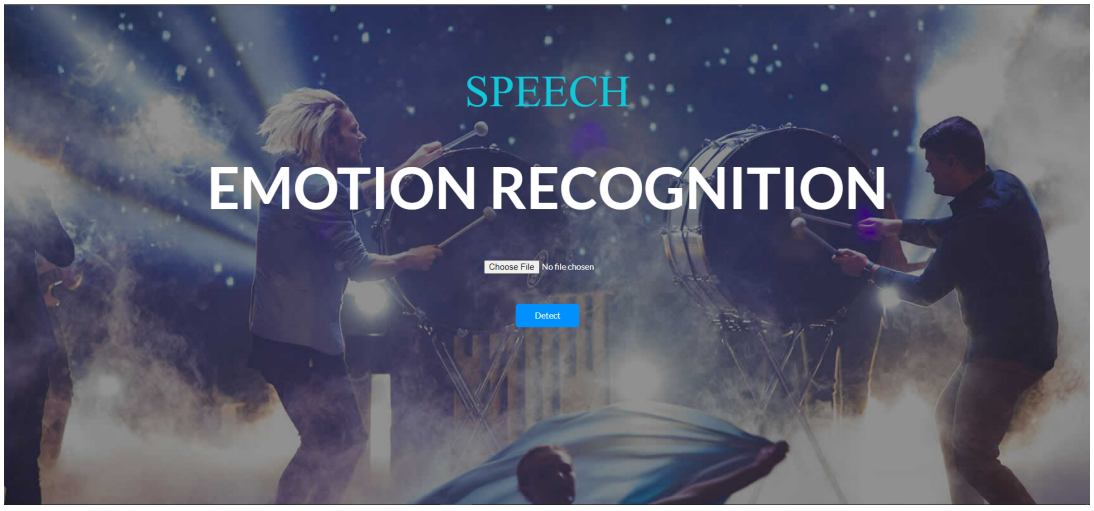
Models	lightgbm	RandomForest	Decison Tree	NaiveByes Classifier	Svc	MLP
Data before Augmentation	0.625000	0.541667	0.277778	0.427083	0.541667	0.569444
Data after Augmentation	0.725694	0.651042	0.246528	0.420139	0.828125	0.812500

With data augmentation, we can see there is a tremendous improvement in the accuracy score of the models, with maximum increment can be seen in MLP Classifier.

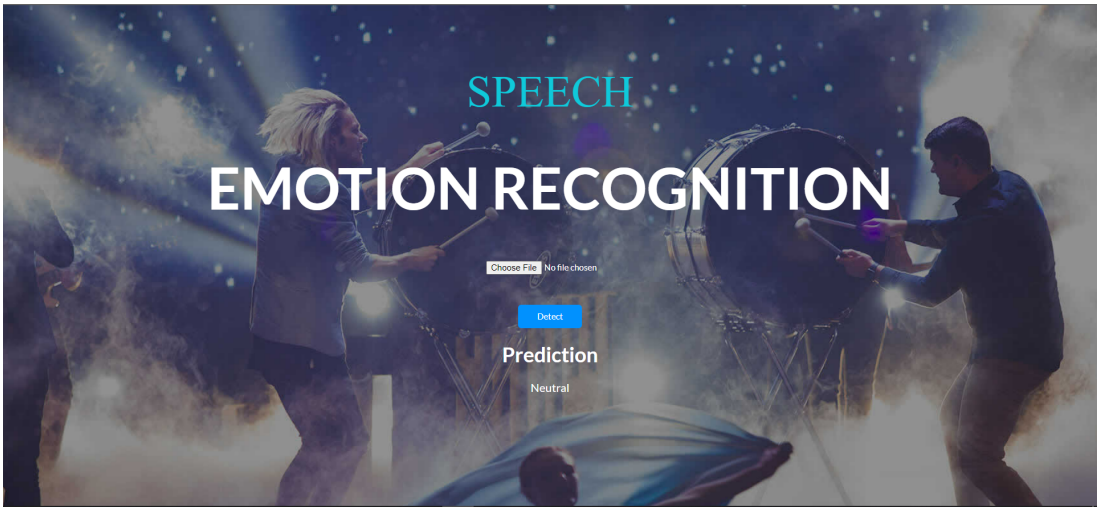
DEPLOYMENT

The model was deployed on the web with the Github code: here
Some screenshots of the same are attached.

Step-1 Click on **Choose File** and select a audio file(.wav file format)



Step-2 Click on the **Detect** button you will get your prediction



CONCLUSION

The graphs plotted and the table shows that models like Light GBM, SVC and MLP have comparatively higher accuracy than the rest. This trend pertains similarly even when the data is augmented with noise. The model was finally deployed on the web using Flask library. The accuracy obtained was 82% on the best model.
The voices can be differentiated from each other using the intensity of the voices which can be clearly seen as different using the spectrogram plots and mfcc values. The trained model is capable of predicting the emotion of the speaker given a sound.

CONTRIBUTION

The learning and understanding of the concepts were done as a team. The individual contributions are given below:

- **Dhruv Mahajan(B20EE016):** Flask model deployment, report ,classification models and comparative analysis
- **Haardik Ravat(B20EE021):** Pre-processing and exploratory analysis,report ,data augmentation and model tuning

REFERENCES

- https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio
- https://librosa.org/doc/latest/index.html
- https://medium.com/@makcedward/data-augmentation-for-audio-76912b01fdf6
- https://www.researchgate.net/publication/335360469_Speech_Emotion_Recognition_Using_Deep_Learning_Techniques_A_Review