Julia Chancey, 6592-4559
CIS4930 Individual Coding Assignment
Spring 2023

1. Problem Statement

Audio Emotion Recognition is a challenging task that can have a large range of applications that can be implemented as real-world solutions. By building a successful recognizer of emotion through audio files, the possibilities are endless – from improving human-computer interaction to enhancing mental health support. The goal of this assignment was to build a machine learning model that could accurately classify audio clips into different emotion categories – angry, fear, happy, and sad. To achieve this, I used the acoustic features extracted from the audio data to train and evaluate the model I implemented - a Random Forest Classifier.

2. Data Preparation

For this assignment, there were several needed steps in order to properly prepare the data. After getting the dataset of audio files with the following emotions: angry, fear, happy, sad, I iterated through them and used the Librosa library to extract Mel-frequency cepstral coefficients (MFCCs) from each audio file. MFCCs are used as features for audio classification because they represent the spectral characteristics of the audio signal.

After extracting the MFCC features, I split the dataset into training and testing sets using the 70:30 split. Then, I scaled the feature matrix to be between -1 and 1 using the MinMaxScaler from the scikit-learn library.

*Afterwards, I concatenated the MFCC features for each audio file into a single feature vector that gave me a feature matrix of size (number of samples, 20 * number of frames). The number of frames for each audio file was determined by dividing the length of the audio signal by the frame length used in the MFCC extraction function.*

Then, I got the feature matrix and used it for training and testing the machine learning model.

3. Model Development

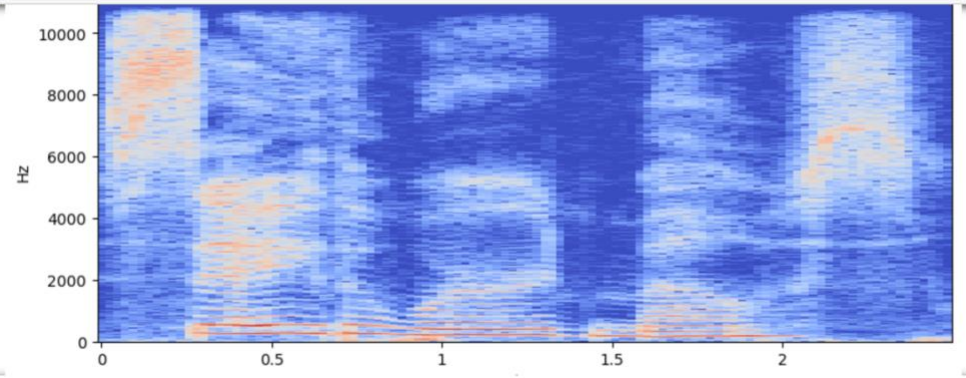
- Model Training

Once I had my training set which was equivalent to 70% of the given audio files, I trained the Random Forest Classifier using the extracted acoustic features from the training dataset. I set specific parameters for my model, such as number of tree, and then used cross-validation and grid search to determine the optimal hyperparameters. Once I found the best hyperparameters, the model was trained for a specified number of epochs (100) to obtain the best accuracy on the testing set.

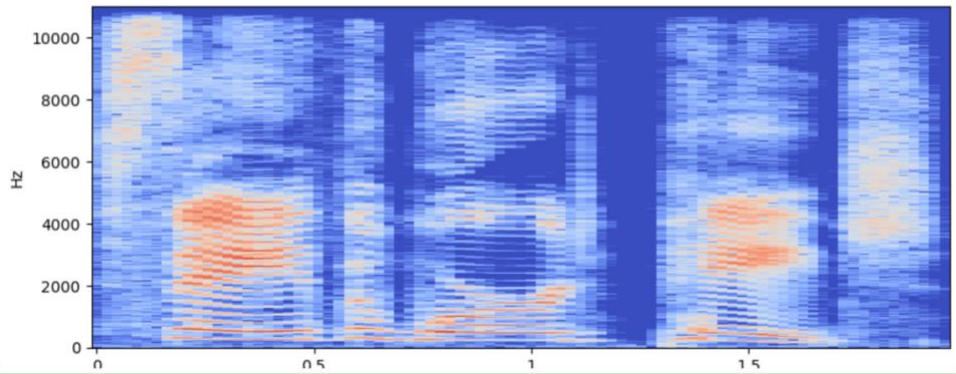
- Model Evaluation

Before getting into the model evaluation, I first plotted the data from some sample audio files that resulted in the following:

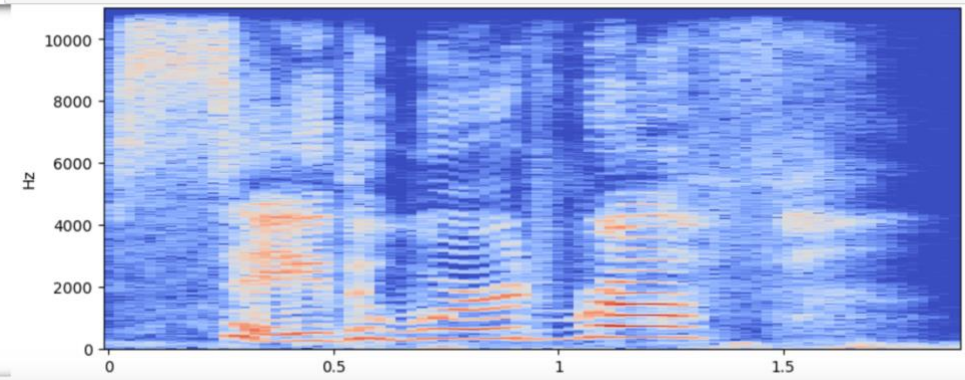
In [80]:
```
plotData(sadTrainSample, sadTrainDir)
```



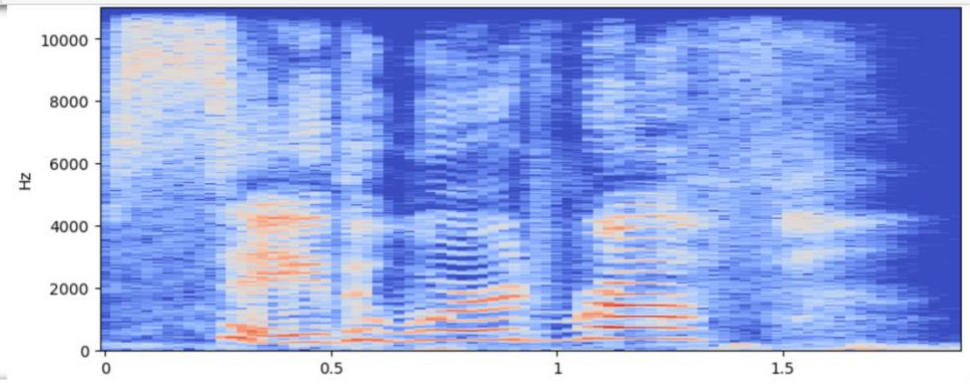
In [77]:
```
plotData(angryTrainSample, angryTrainDir)
```



In [78]:
```
plotData(fearTrainSample, fearTrainDir)
```



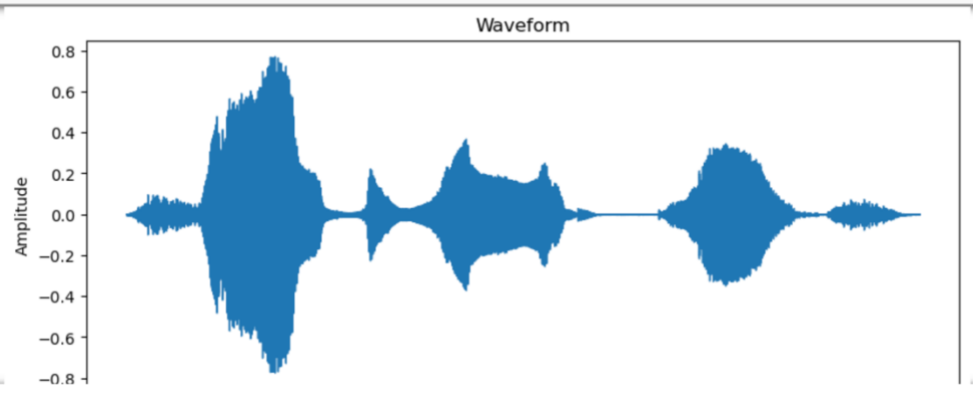
```
In [79]: plotData(happyTrainSample, happyTrainDir)
```



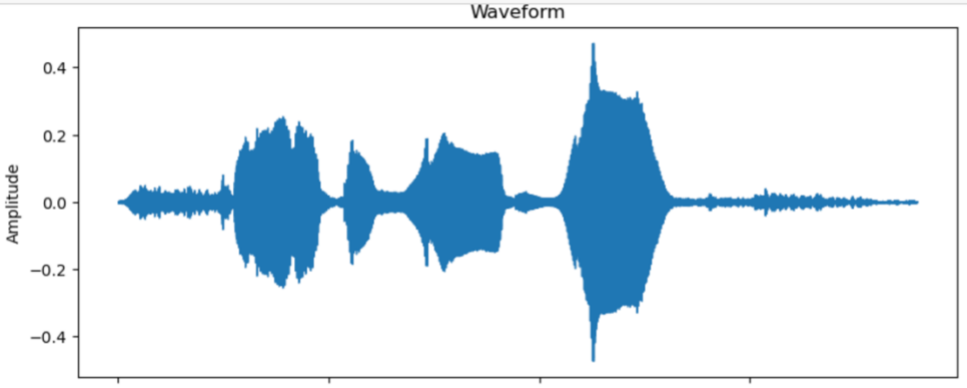
The above images are a spectrograms of some sad, angry, fear, and happy audio files, respectively.

Additionally, I also plotted the waveform of sample audio files for each emotion shown as followed:

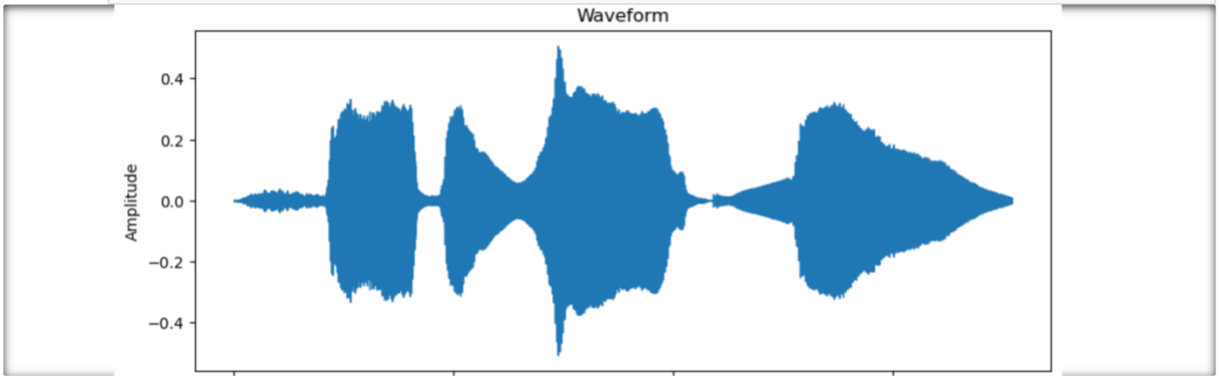
```
In [84]: plotAmplitude(angryTrainSample, angryTrainDir)
```



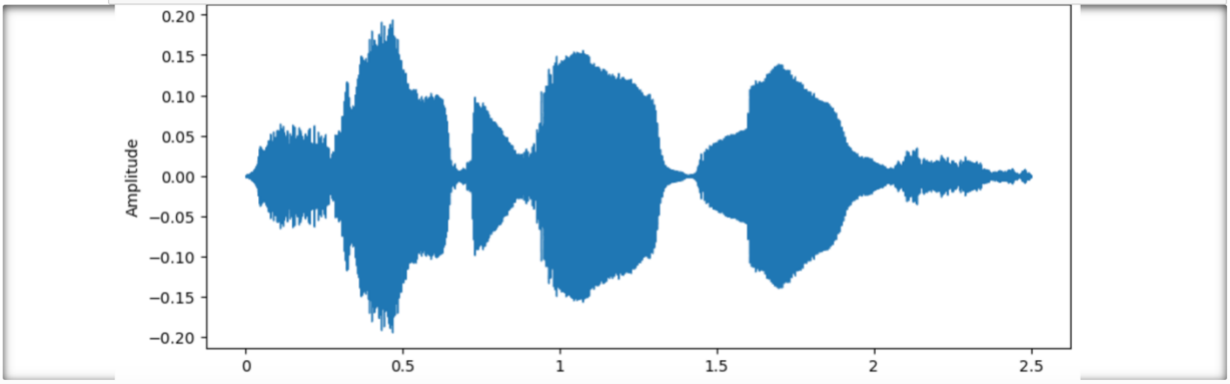

```
In [85]: plotAmplitude(fearTrainSample, fearTrainDir)
```



```
In [87]: plotAmplitude(happyTrainSample, happyTrainDir)
```




```
In [88]: plotAmplitude(sadTrainSample, sadTrainDir)
```



These are amplitude plots for five sample audio files from each emotion – angry, fear, happy, sad (respectively) that helped me distinguish different characteristics from each emotion and enlightened me on how an ML classification model may be able to accurately predict which emotion is being displayed in each audio clip.

After preparing the data, splitting the sets up, and training the model, I printed out the accuracy of my Random Forest Classifier which outputted the following:

```
# Compute accuracy score
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy}')
```

```
Accuracy: 1.0
```

My findings were an accuracy of 1.0 which I'll discuss in the following section of the report.

4. Discussion

The accuracy I reached with the Random Forest Classifier was 1.00 (which means an 100% success rate), meaning the model is able to correctly predict the emotions of all audio samples in the testing dataset. However, given the very high accuracy on the testing dataset, I am suspicious that my model may have been overfit to the training data or that the testing dataset was too small or not truly representative of the true distribution of the data. Based on the given accuracy result, I could conclude that my model performs well enough and can serve as an audio emotion

recognizer. However, there is a small possibility of the accuracy decreasing with other data which I won't be able to find out about until new audio clips are used.

Some challenges I faced throughout the development process were feature extraction, overfitting, and model selection. When I was trying to select the appropriate features, I was unsure of which ones to choose because I was worried about representing the emotions in the audio files accurately given that this selection is crucial for the success of the ML model. Another challenge was my worry of overfitting. It could be that my model fits the training data too closely and fails to generalize to new data but I won't be able to find that out until my model is tested with new audio samples. Lastly, choosing the right model was definitely hard because I wanted to make sure I chose the right model architecture for the problem at hand that would also perform the best. To solve my challenges, I visualized my data, tuned my hyperparameters, and cross-validated throughout my project's development.

5. Appendix

https://github.com/julachancey/cis4930-hw3.git
This is the github link to my code that used a jupyter notebook and python.