Cole Slatt 6/28/2020

Capstone Project Technical Report

As we've sought to endow computers with human-like abilities, it's been a priority to be able to communicate with them as we do with each other. There's been significant emphasis in recent years on using computers to understand human language, but there's much more to verbal communication than words and grammar; the tone of our voice embeds a great deal of information, particularly related to how we're feeling. If we want to create technologies that seamlessly integrate into our lives, they should be able to understand and respond to the auditory cues that we use to express our emotional and mental states with each other.

The question I asked at the beginning of this project was whether we can use data science and machine learning to classify emotions based on the sound of someone's voice. This seems to be an emerging area of focus; most of the labelled datasets I found were published in the last five years. At the same time, the interest in this topic has been explosive; a Google search of 'emotional audio classification' will yield many recent projects that attempt to tackle the problem. In a particular project published on Kaggle, only convolutional neural networks were used, with some questionable preprocessing steps taken. I also found a research paper discussing cutting edge techniques that made use of multiple recurrent neural networks, and aggregations of multiple audio based features for the purposes of audio based classification. The strategies deployed to tackle this problem seem to be varied and without many hard and fast rules. I am confident that my approach to this problem is unique and I hope others can learn from my findings.

I made use of two datasets for the training and testing of models, namely the Toronto emotional speech set (TESS), and the Surrey audio-visual expressed emotion database (SAVEE). TESS is comprised of 2800 recordings of two actresses saying targeted words, all labelled as one of seven emotions; anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral. The file structure is as follows:

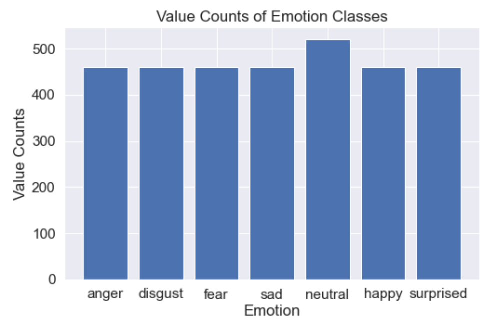
```
— TESS
  ├── OAF_angry
  ├── OAF_disgust
  ├── OAF_fear
  ├── OAF_happy
  ├── OAF_neutral
  ├── OAF_pleasant_surprised
  ├── OAF_sad
  ├── YAF_angry
  ├── YAF_disgust
  ├── YAF_fear
  ├── YAF_happy
  ├── YAF_neutral
  ├── YAF_pleasant_surprised
  └── YAF_sad
```

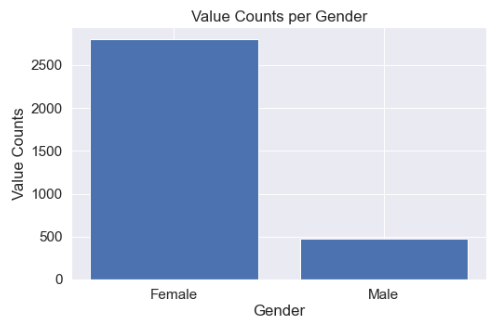
Each folder in these directories contains 200 recordings encoded as .wav files, with each recording containing a phrase with one of the 200 targeted words. The SAVEE dataset consists of 480 recordings of four male speakers, labelled by the same emotions as TESS. The directory of files is organized as follows:

```
- SAVEE
  ├── DC
  ├── Info.txt
  ├── JE
  ├── JK
  └── KL
```

Each folder in this directory contains 120 recordings also encoded as .wav files, with 30 of those recordings being of class 'neutral', and all other emotional classes having 15 files apiece. All datasets were found available to download for free online.

Preprocessing started with building a data frame consisting of basic information about each file, including the filename, dataset, speaker's gender, speaker's name, duration, and emotional class of each file. From this data frame I was able to reference the actual audio files as needed without having to alter the original file structure whatsoever. I then balanced the data by emotional classes and gender. This reduced the number of files in my data frame from 3280 files to 840.

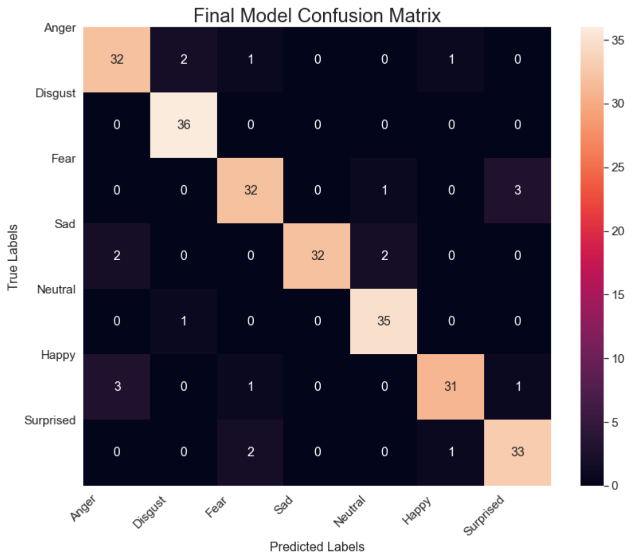




The first feature that I extracted from the audio files were the mel frequency cepstral coefficients (MFCCs). This is one of the most commonly used features in speech processing, as it maps the measured frequency content of audio to a scaled based on observed psycho-acoustic phenomena, related to how humans perceive pitch and loudness. You can specify the number of coefficients and the size of frames that each set of coefficients is based on. More coefficients and smaller frames equate to more granular data. Common practice is to use 13-20 coefficients, and some people have taken the moving average of those coefficients as a primary feature. With some experimentation and intuition, I found that taking 40 MFCCs, each averaged over time, minimized information loss while expressing the data in a concise format. This meant that there were initially 40 features per file. I also extracted the unaveraged 40 MFCCs, resulting in a 2D array for each file. I took the mean and total gammatone frequency cepstral coefficients (GFCCs), which are based on another psychoacoustic scale. I also used the raw audio data resampled at 11 kHz. While audio is commonly sampled at 41 kHz, the bulk of frequency content in speech can be captured with a sampling rate of 11 kHz, therefore minimizing the number of columns. All 1D features were scaled with a standard scaler before modelling.

As this is a classification problem, classification accuracy was the primary metric used to assess model performance. The first models I tried were logistic regression, linear, kernel, and bagged support vector machines, and random forest, all trained and tested with 40 MFCCs per file. The top performing of these was the RBF kernel SVM which classified emotion with 88% accuracy on the test set. I then went on to train a dense neural network and a 1D CNN on same data. The way that I'd averaged the MFCCs removed any sort of sequence dependancy from the data. I therefore concluded that a dense neural network would be the best choice in this case, but wanted to test a CNN as well to compare performance. The dense neural net performed with 88.8% accuracy, while the CNN performed with 83.7% accuracy. The next model I tried was a 2D CNN trained on the full length of MFCCs, meaning there were sequence dependancies within each file's 2D array. This model classified emotion with 82.9% accuracy. Following this I tried a series of models and techniques that were valuable learning experiences, but were not very successful. Using the raw audio of each file I trained a 1D CNN, which initially performed with just 53.7% accuracy and took a long time to train. To increase performance I augmented the data by applying various audio transformations to each file, including pitch shifting, time stretching, adding noise and randomizing the placement of padding. This got the model's accuracy up to 65%. I also spent a great deal of time trying to work with recurrent neural networks. I initially tried to train one with the same augmented data as I'd used with the CNNs, but the number of columns made it prohibitively slow to train. After this I performed four major experiments with RNNs; I sliced each audio file in the augmented data into ten pieces and re-added each piece onto the end of the data frame; I downscaled each audio file by pitch shifting so that they were all the same length without needing to pad the audio; I tried training an RNN with the moving average of 40 MFCCs; and finally I encoded the MFCCs with random tree embedding in order to create a fixed length vocabulary from which an RNN would be more easily able to pick up regular patterns in the data. It did not seem like an RNN was able to train effectively on any of this data. The last series of models I tried were all based on 40 GFCCs. With these I trained a RBF kernel SVM, a dense neural net, and a 2D CNN, which had classification accuracies of 82.5%, 86.9%, and 82.5%, respectively.

After collecting the prediction probabilities from each model, I wrote a function to test which combinations of predictions would score the highest accuracy, combining predictions through multiplicative probabilistic voting. The best models ended up all being based on MFCCs, and included the 1D CNN, 2D CNN, dense neural net, random forest, and RBF kernel SVM. The combination of these models yielded a 91.7% final accuracy score, and a confusion matrix as follows:



True \ Predicted	Anger	Disgust	Fear	Sad	Neutral	Happy	Surprised
Anger	32	2	1	0	0	1	0
Disgust	0	36	0	0	0	0	0
Fear	0	0	32	0	1	0	3
Sad	2	0	0	32	2	0	0
Neutral	0	1	0	0	35	0	0
Happy	3	0	1	0	0	31	1
Surprised	0	0	2	0	0	1	33

This level of accuracy shows how possible it is to teach machines how to recognize emotions through speech. My analysis has really proven how important the MFCCs are in application of this problem, and more generally how much feature selection and preprocessing can impact model performance. There are numerous practical applications for this project; we could build tools that can diagnose and track mood and mental health disorders, assess the level of customer engagement with call centres, and help voice assistant services like Siri and Alexa respond to how we're feeling. In future steps with this project, I would really like to figure out how to implement RNNs, as they should be a very powerful tool for working with audio. I'd like to train these models on significantly more data to improve accuracy and increase robustness, and I'd like to make use of other audio based features. Furthermore I'd like to experiment with more sophisticated kinds of audio transformations for data augmentation, to improve the models ability to classify recordings captured in varied acoustical environments.