Classifying musical Instruments *Preston Raab*
8.5/10

1. Description of the project
Gathering and classifying audio data is important in many fields. Apps like Shazam use audio to determine what song is playing, wildlife researchers process audio to identify where animals are, and smart home systems like Alexa use voice data to determine if a person it speaking to the device. Since audio is a wave, audio data shares characteristics with measurements from gravity waves, earthquakes, and photodiodes. **As a simple project, I propose a musical instrument classifier**. Given a recording of (at most) a single instrument, it will be able to determine what instrument is playing, or if no instrument is in the recording. I think we could additionally label the data with "Woodwind", "Brass", and "Vocal", so that if a novel instrument is heard, we could still have a means of classifying it.

2. How and from where would the data set be gathered and labeled
We would record people playing various musical instruments on a laptop, or a phone. We would name the large files with the appropriate labels. For example: Sarah.flute.woodwind.mp3, or birds.no_instrument.mp3. After transferring the file to a computer, we would use a python script to convert the file to a standard .wav format, with a consistent sampling rate, then split it into 5-second chunks. That script would need to be run on any novel future input.

Each piece of test/training data would then be a 5-second wav file, sampled at a consistent rate, with the labels included in the name of the file.

3. What features the data set would include
We would need to do some processing on the data to get the right features. I know that the most important feature set would be a frequency spectrogram, which we could get from any short snippet of audio, say 0.1 seconds. The problem I see is that taking every 0.1 second snippet, finding the spectrogram, and then using every spectrogram to train a model would be far too many inputs. We can't train a model on that many inputs. On the flip side, labelling every 0.1 second spectrogram as belonging to the target instrument would be incorrect, since the player potentially needs to breathe.

One solution would be to only label samples above a certain decibel level, and discard the rest. I think this would be better than training another model to learn the difference between sound and silence.

Our model would label the remaining short snippets using multi-hot encoding (since an instrument can be both a flute and a woodwind), and yield an ultimate probability distribution by averaging the short sample outputs. I think we would ignore that overall probability when training the model though, and stick with the outputs from the short samples.

To summarize, the input features that are entered into the model is a frequency spectrogram, and our target features would look like this:

Not Music	Violin	Strings	Harp	Trumpet	Brass	Voice	Piano	Guitar	Organ
0	0	1	0	0	0	0	0	1	0