

Capstone Project #2 – Milestone Report

Jupyter Notebook: [Link](#)

The Problem

The project will attempt to classify sound files of less than four seconds each into 10 classes of urban sounds, such as a dog barking or a car horn.

The Client

The fictional client is a home security company that wishes to build a more effective system for their clients that helps to anticipate problems in an attempt to prevent problems before they happen. By analyzing the ambient noise surrounding the house, they hope to identify sounds that could cause a potential threat. Their system can then be programmed to take an appropriate action for these threats, such as trigger an alarm or lighting system, notify the homeowner or alert the appropriate authorities.

The Data

The data was taken from the Analytics Vidhya website.

<https://datahack.analyticsvidhya.com/contest/practice-problem-urban-sound-classification/>

The data consists of 5435 .wav files of no more than four seconds in length. Each of these sounds is labeled with one of ten possible urban sounds.

The Approach

An initial deep learning model will be fit to the data. After that, experimentation with the model parameters will be done to find the model with the highest accuracy.

Exploratory Data Analysis

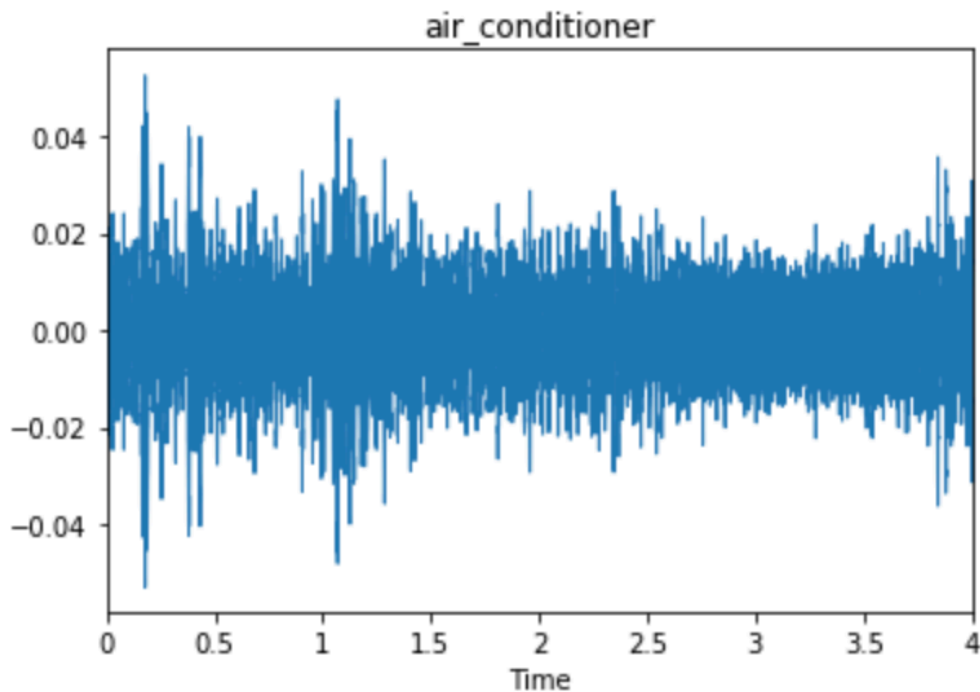
The data consists of 10 different types of urban sounds: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gunshot, jackhammer, siren and street music. The breakdown of the number of data points for each type of sound is shown below:

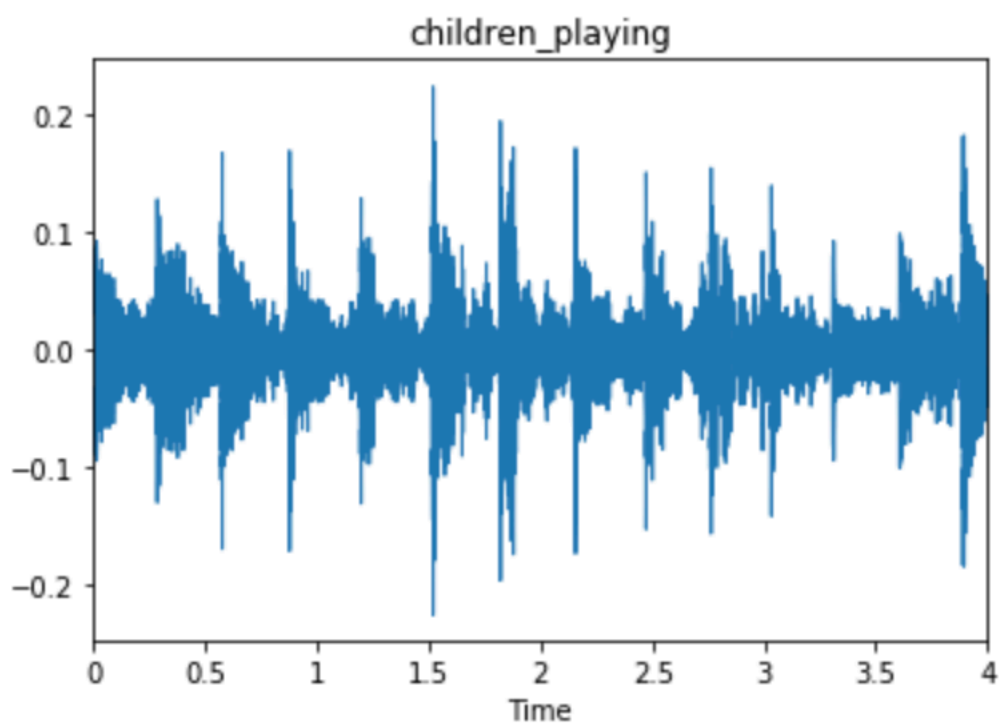
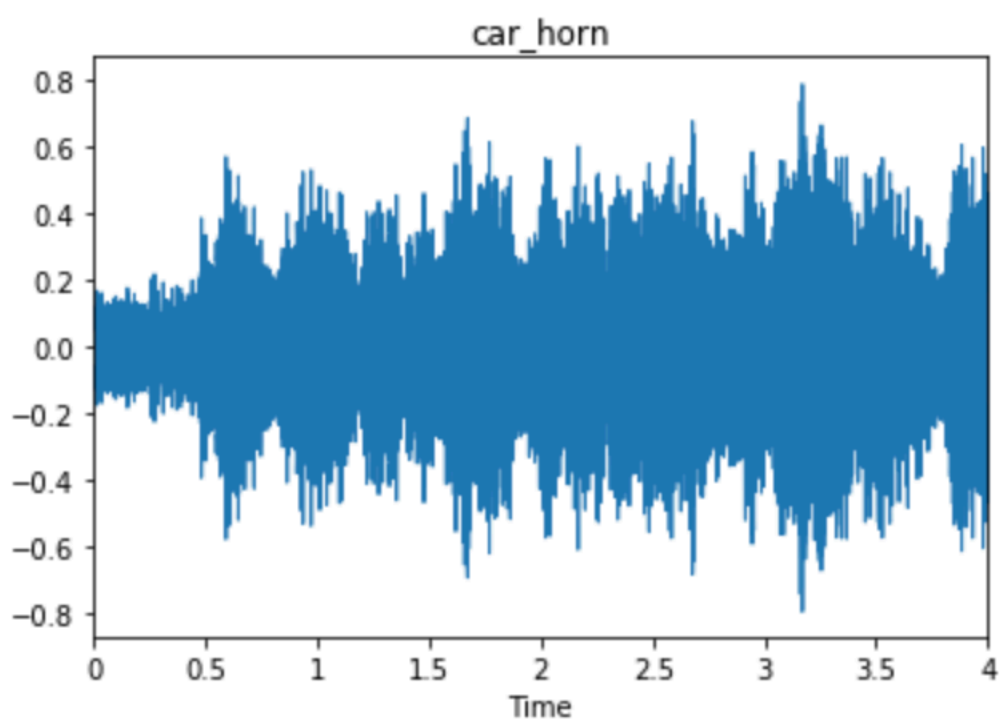
jackhammer	668
engine_idling	624
siren	607
street_music	600
dog_bark	600
air_conditioner	600
drilling	600
children_playing	600
car_horn	306
gun_shot	230

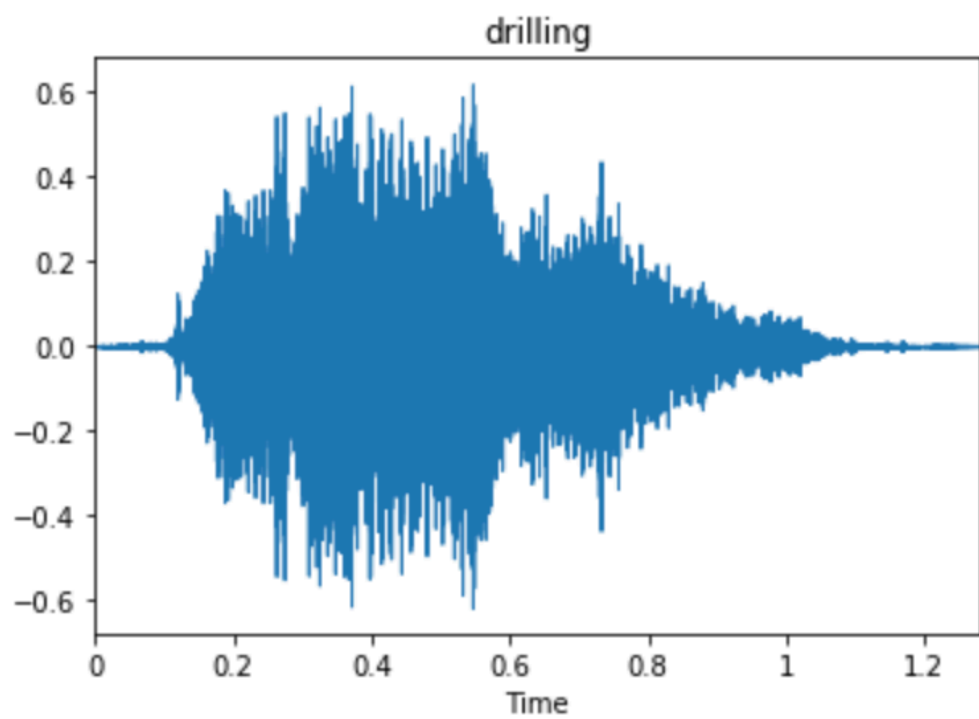
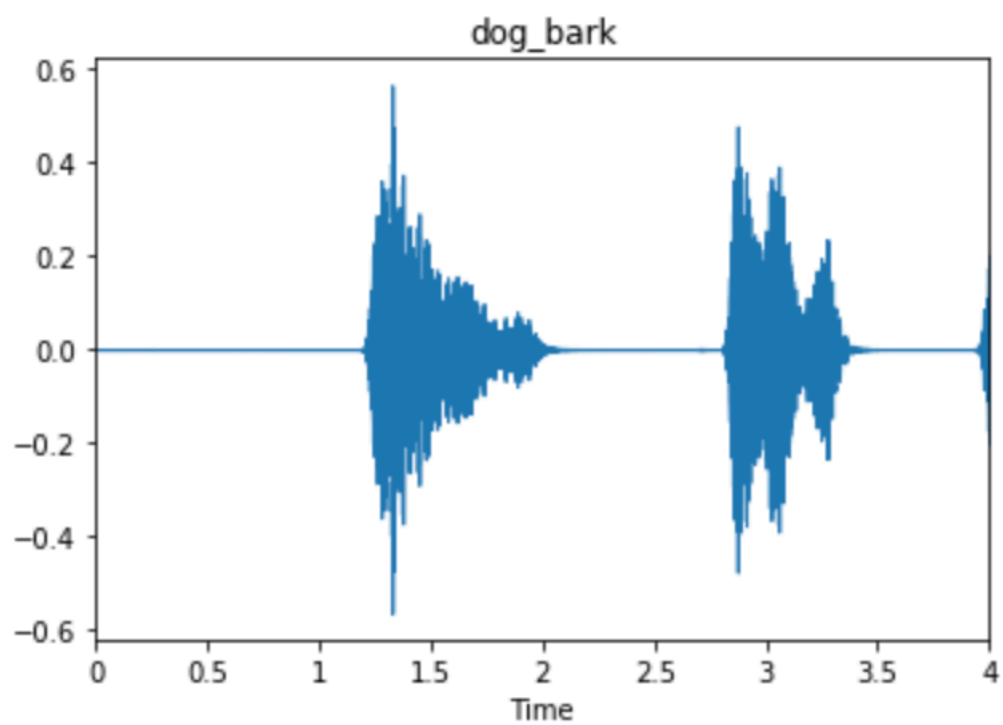
Name: Class, dtype: int64

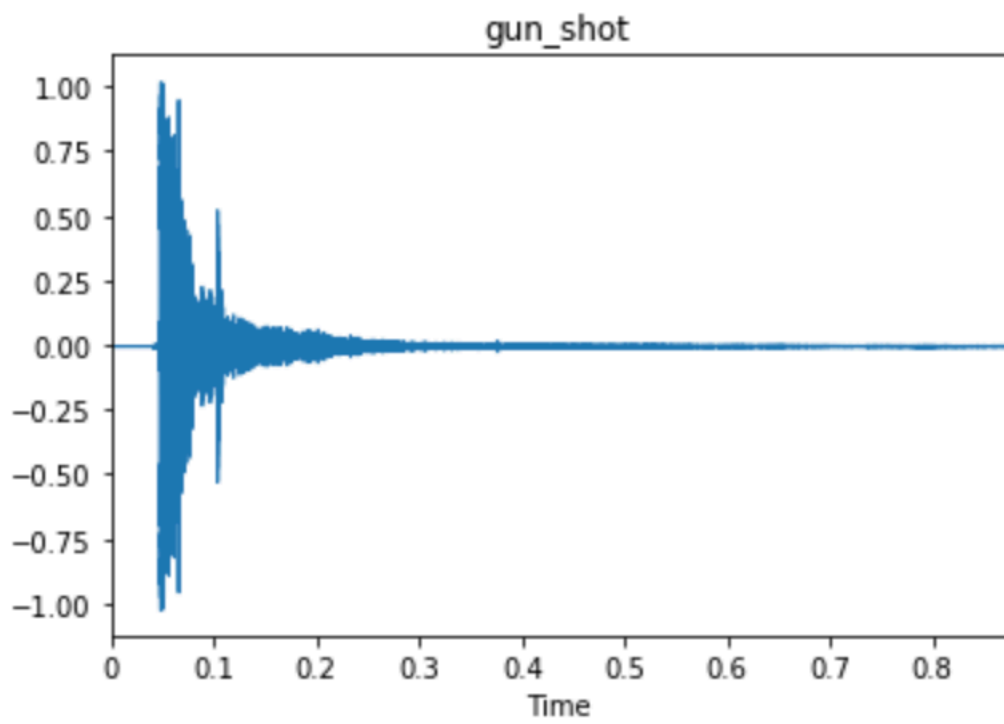
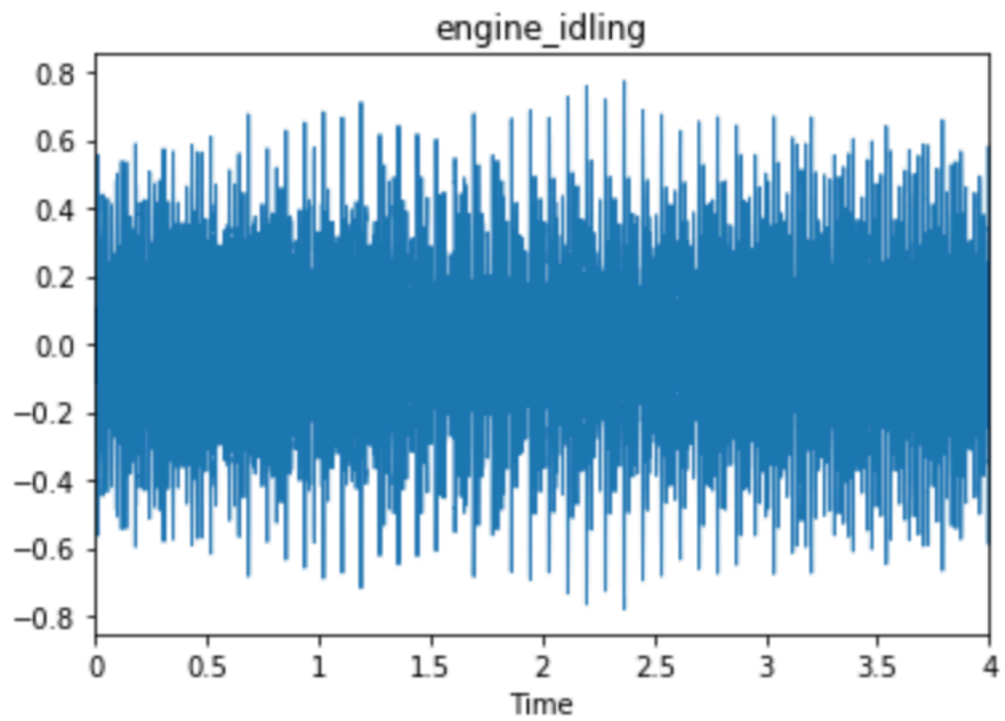
It can be seen that for eight out of the ten sounds there are roughly 600 samples. For car horn and gunshot, there are about a half to a third of that number. Moving forward in the analysis, we will use sampling techniques to ensure good representation in our training set for each of the ten classes.

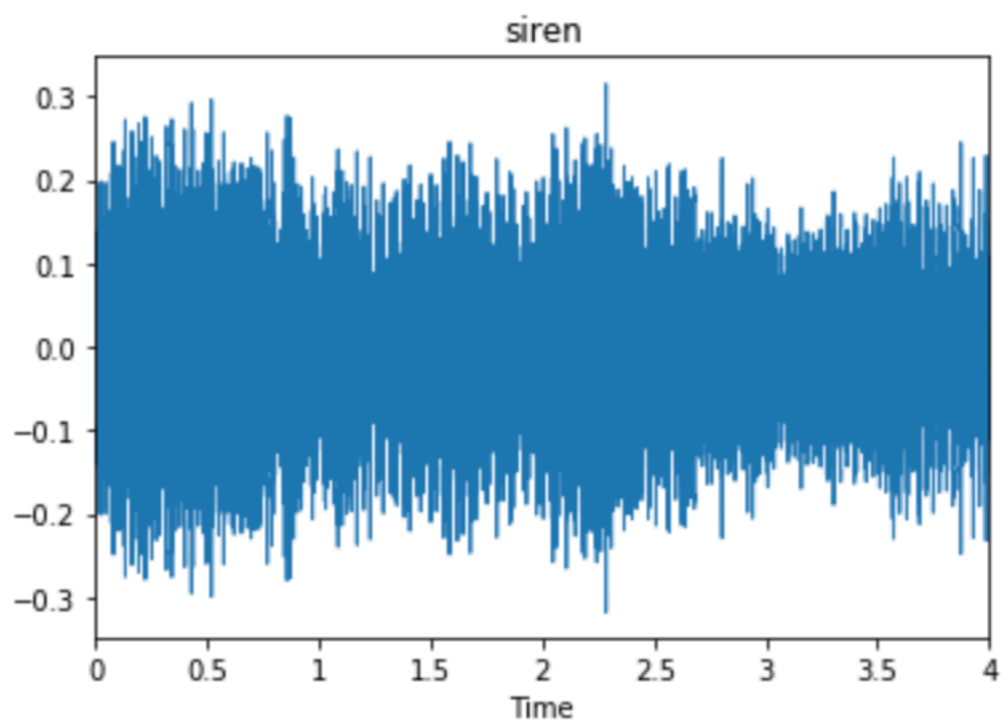
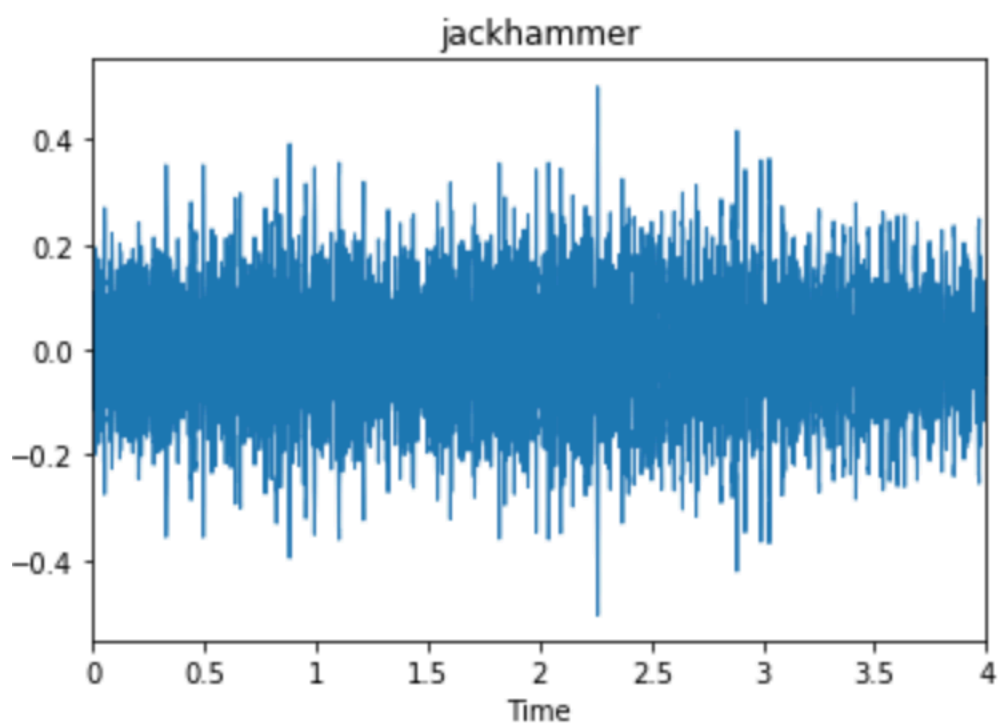
For each of the ten classes, three random files were selected to examine their wave plots. Examples for each class can be seen below. All of the wave plots can be seen in the linked Jupyter notebook.

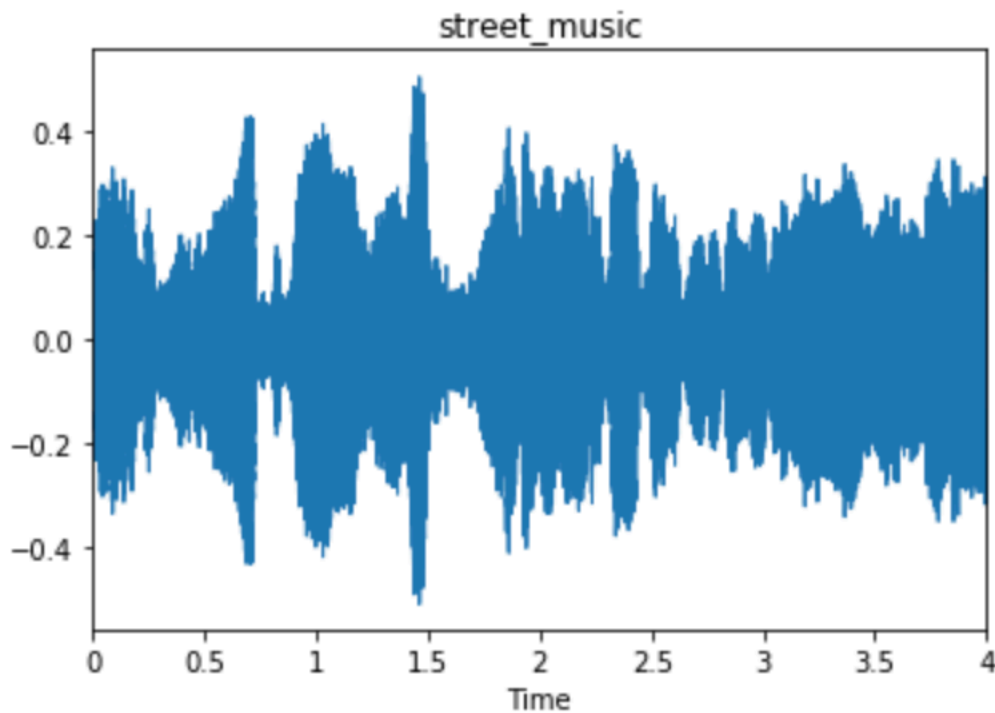












Some observations and trends seen from the wave plots:

- Air conditioner tends to show solid bands of sound with occasional brief spikes. There was also consistency seen in the values of the y-axis.
- Car horn also shows solid bands of sound with the periods of intensity lasting longer than those for air conditioner.
- Children playing showed wide variation between the three plots. These may be difficult to classify other than their dissimilarity to the other categories.
- Dog bark was characterized by short outbursts of intensity surrounded by periods of relative calm.
- Drilling shows a consistent band of noise but also show signs of well-defined start and/or end points.
- Engine idling shows consistent bands of sound throughout the clip but also tended to have a rhythmic quality with repetitive trends.
- There was inconsistency in the gunshot plots. Two of them show evidence of what would likely be expected: a loud burst of sound followed by little noise. The third plot shows a large band of sound throughout, maybe due to background noise.
- Jackhammer also shows the rhythmic quality seen in engine idling but with tighter bands.
- Siren sounds were solid bands of noise throughout the clip, some showing peak while others were a wall of sound. There is also wide variation in y-axis values.

- Street music shows inconsistent patterns throughout the length of the clip with many peaks and valleys of varying length.

In all, there seem to be differences in the clips that our model will hopefully be able to pick up. Some of the clips that show wide variation within the category may be harder to classify, such as children playing and siren.

Extracting numerical data

In order to extract numerical data from the sound files for input into the neural network, the Python package *librosa* was used. The following features were extracted.

- Mel-frequency cepstrum coefficients: A representation of the short-term power spectrum of a sound.
- Chromagram: A 12-element feature vector indicating how much energy of each pitch class is present in the signal.
- Mel-spectrogram: An acoustic time-frequency representation of a sound.
- Spectral contrast: Metric that considers the spectral peak, spectral valley and their difference in each frequency subband.
- Tonnetz: The tonal centroid features

Each audio file will have this data extracted resulting in a total of 193 features for each sound. Note that there were four files that were corrupted (300, 1182, 1488, 3806) so these will be dropped from the dataset and not used in training the model.