Capstone Project #2 – Final Report

Jupyter Notebook: Link
Presentation Slides: 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.

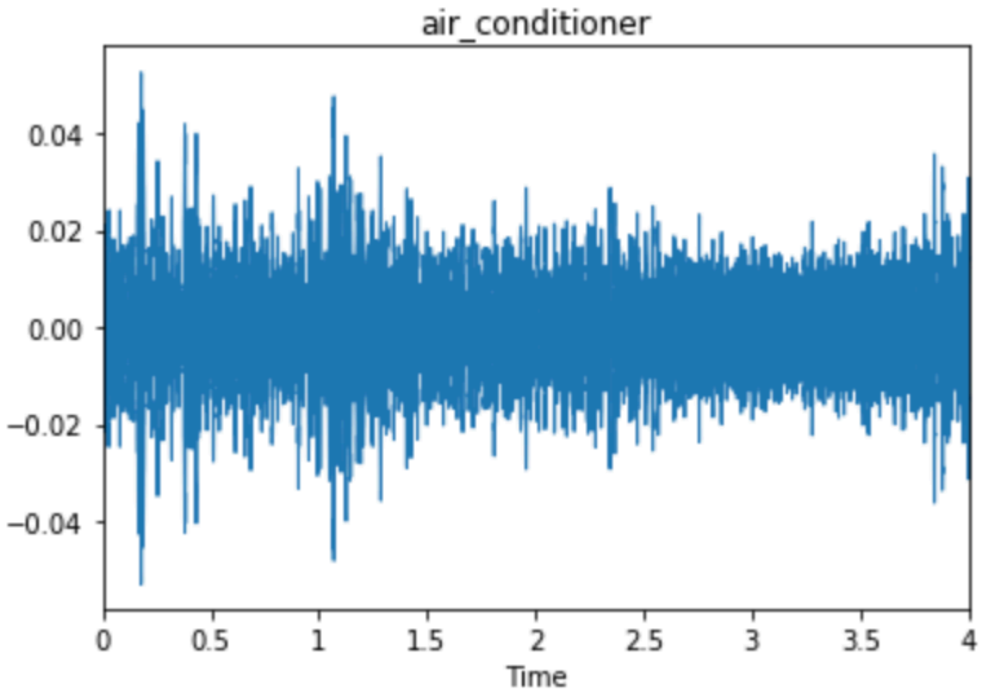
Exploratory Data Analysis

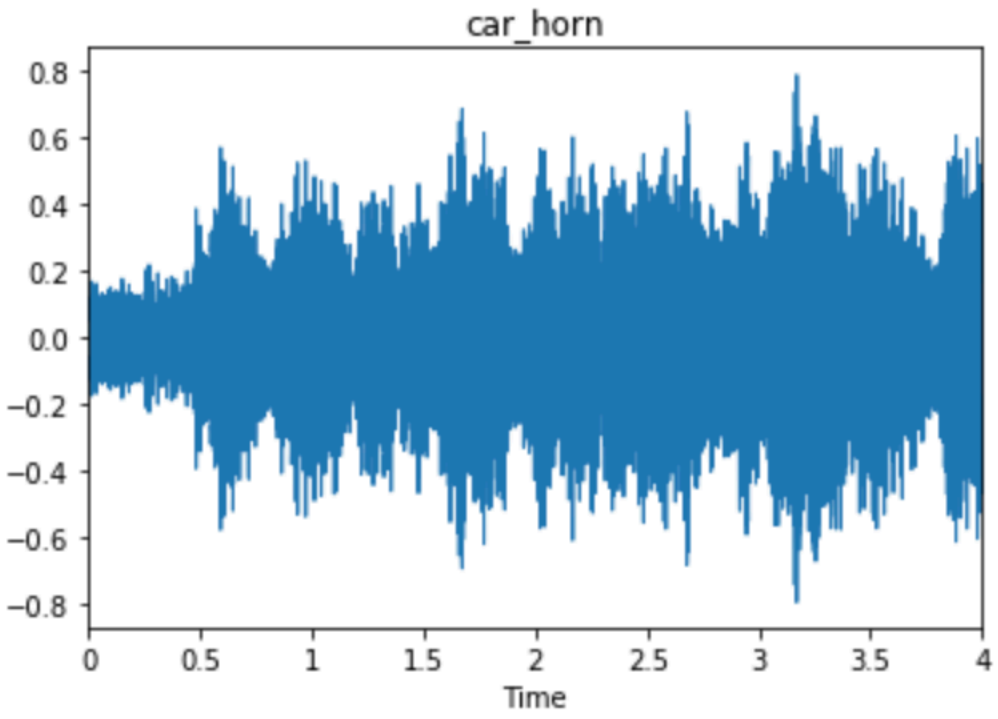
The data consists of 10 different types of urban sounds: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gunshot, jackhammer, siren and street music. The breakdown of the number of data points for each type of sound is shown below:

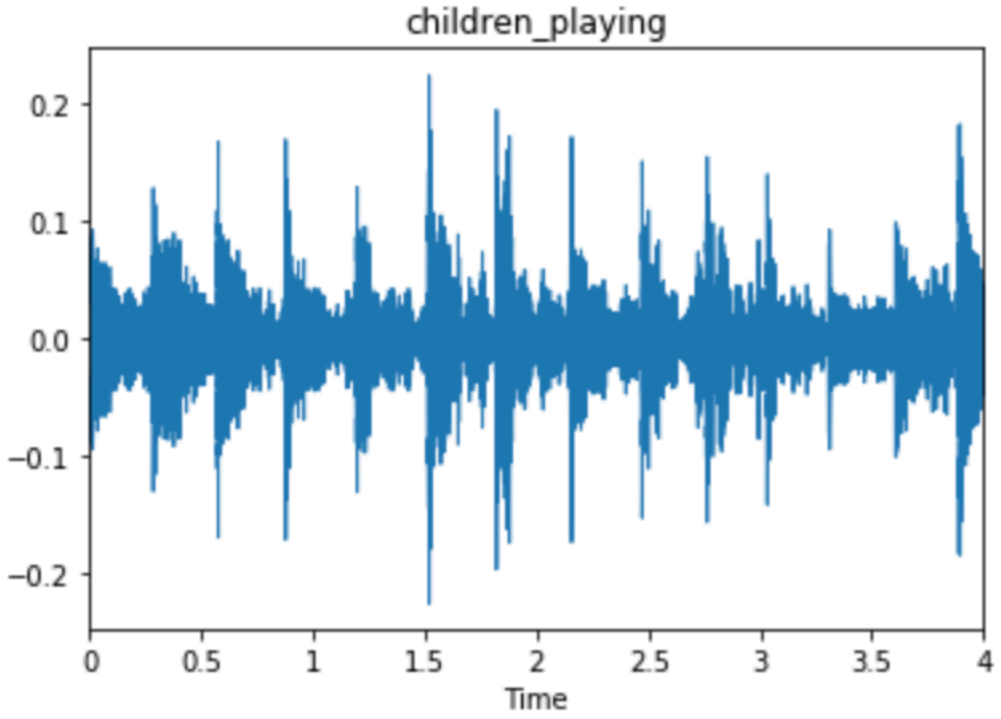
```
jackhammer          668
engine_idling       624
siren               607
street_music        600
dog_bark            600
air_conditioner     600
drilling            600
children_playing    600
car_horn            306
gun_shot            230
Name: Class, dtype: int64
```

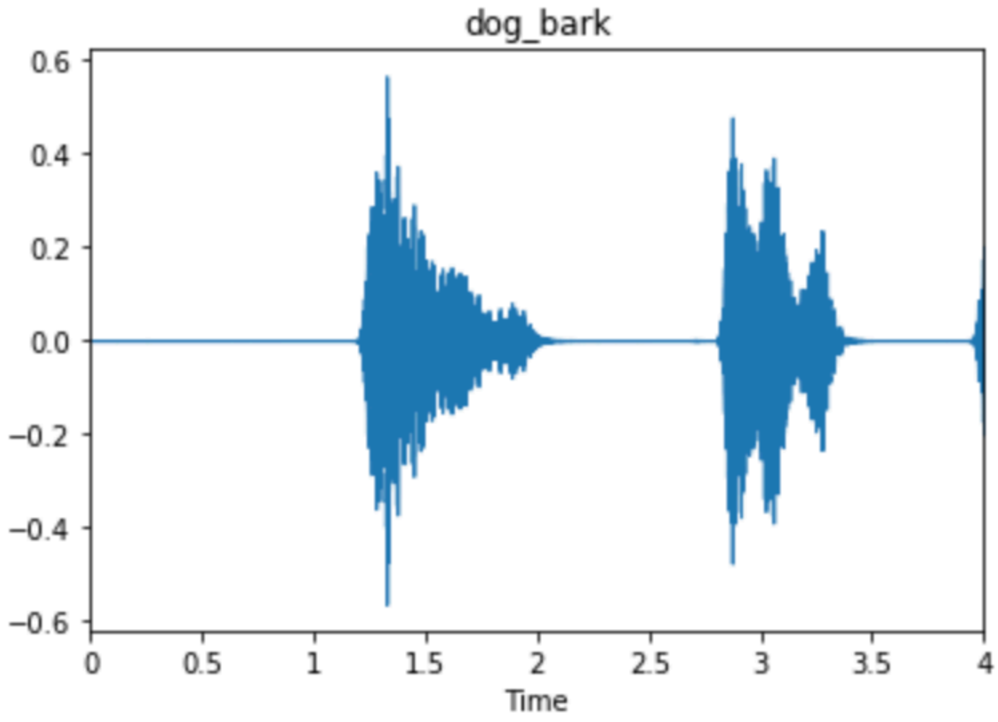
It can be seen that for eight out of the ten sounds there are roughly 600 samples. For car horn and gunshot, there are about a half to a third of that number. Moving forward in the analysis, we will use sampling techniques to ensure good representation in our training set for each of the ten classes.

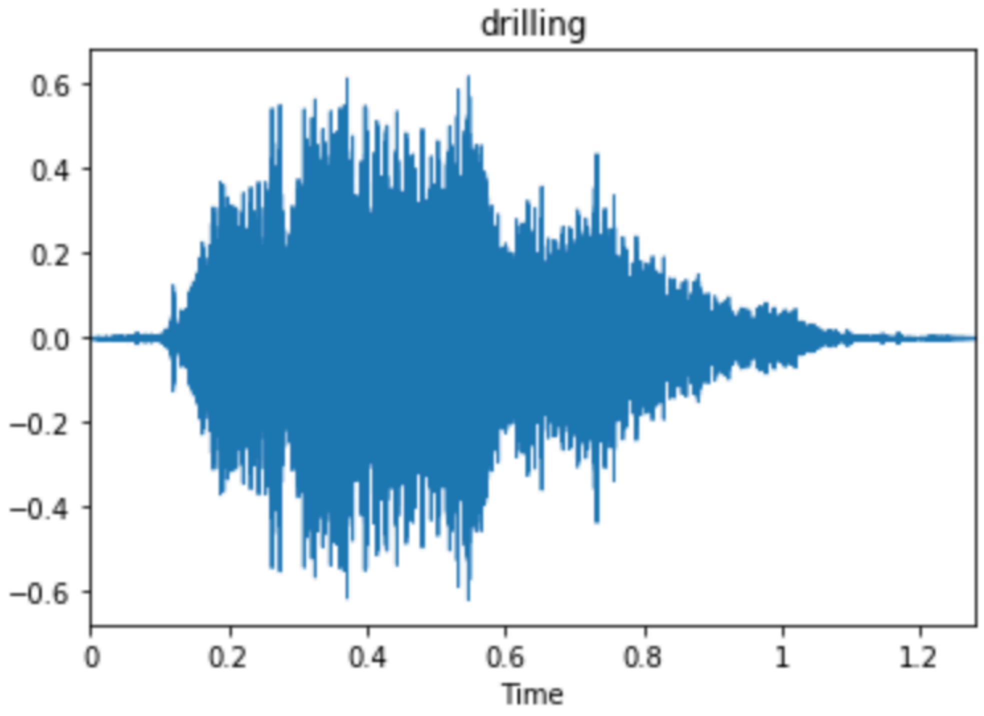
For each of the ten classes, three random files were selected to examine their wave plots. Examples for each class can be seen below. All of the wave plots can be seen in the linked Jupyter notebook.

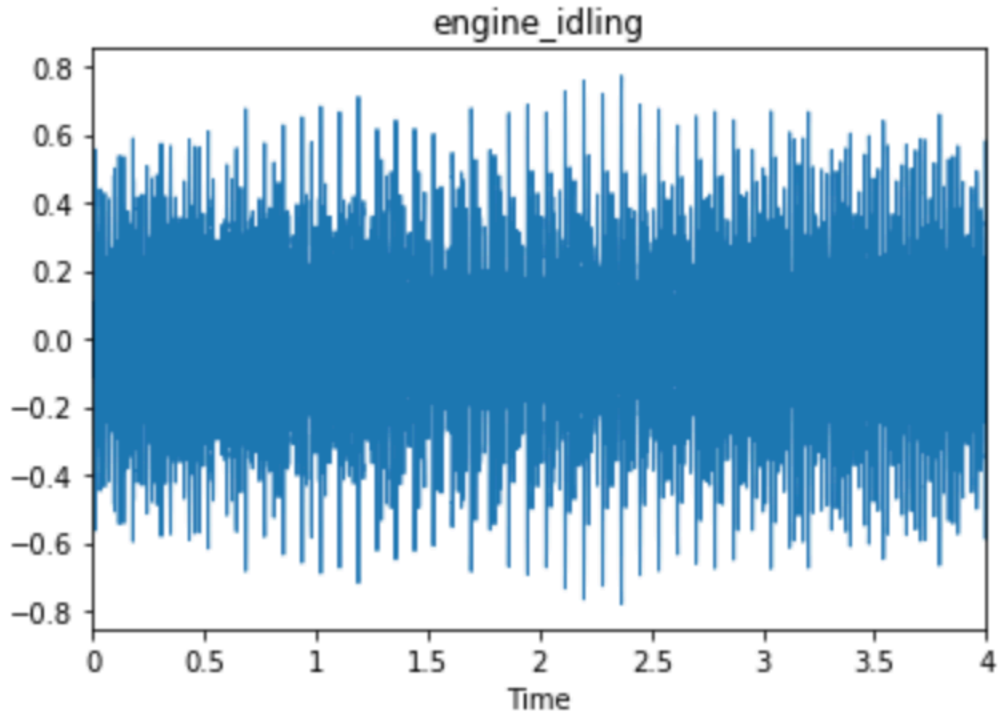


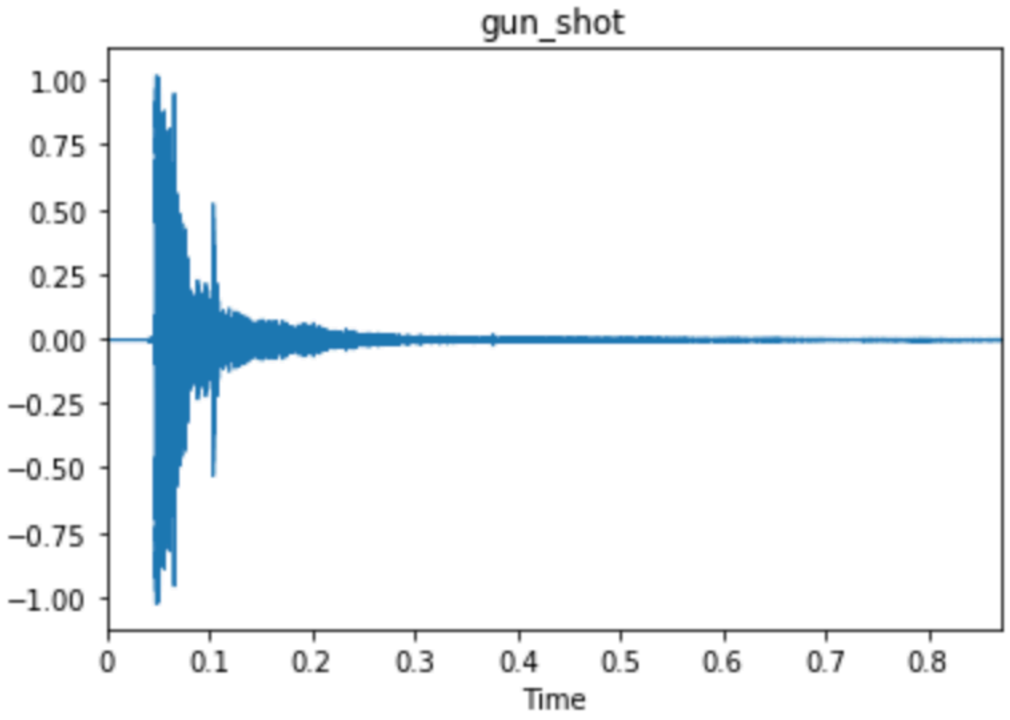

car_horn
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
0
0.5
1
1.5
2
2.5
3
3.5
4
Time

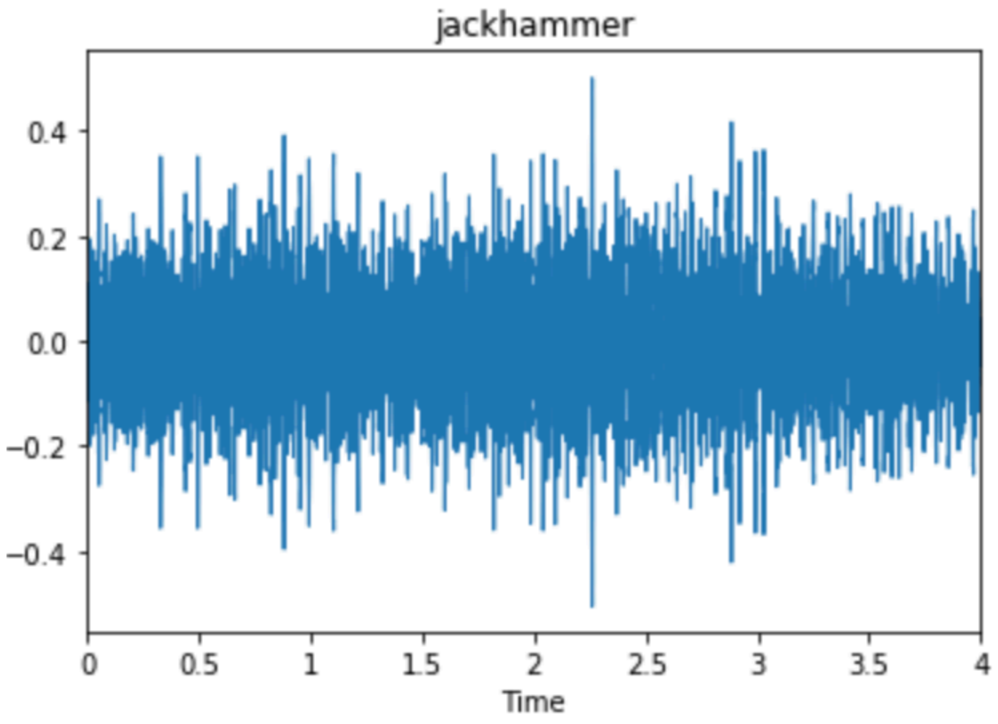

children_playing
0.2
0.1
0.0
−0.1
−0.2
0
0.5
1
1.5
2
2.5
3
3.5
4
Time

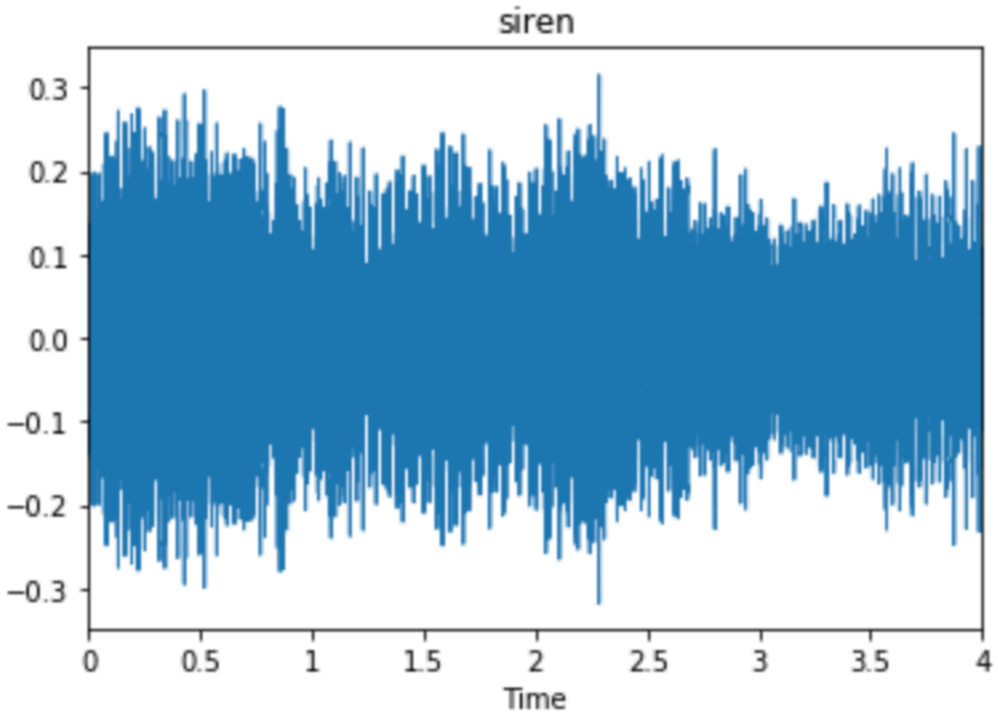

dog_bark
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0
0.5
1
1.5
2
2.5
3
3.5
4
Time

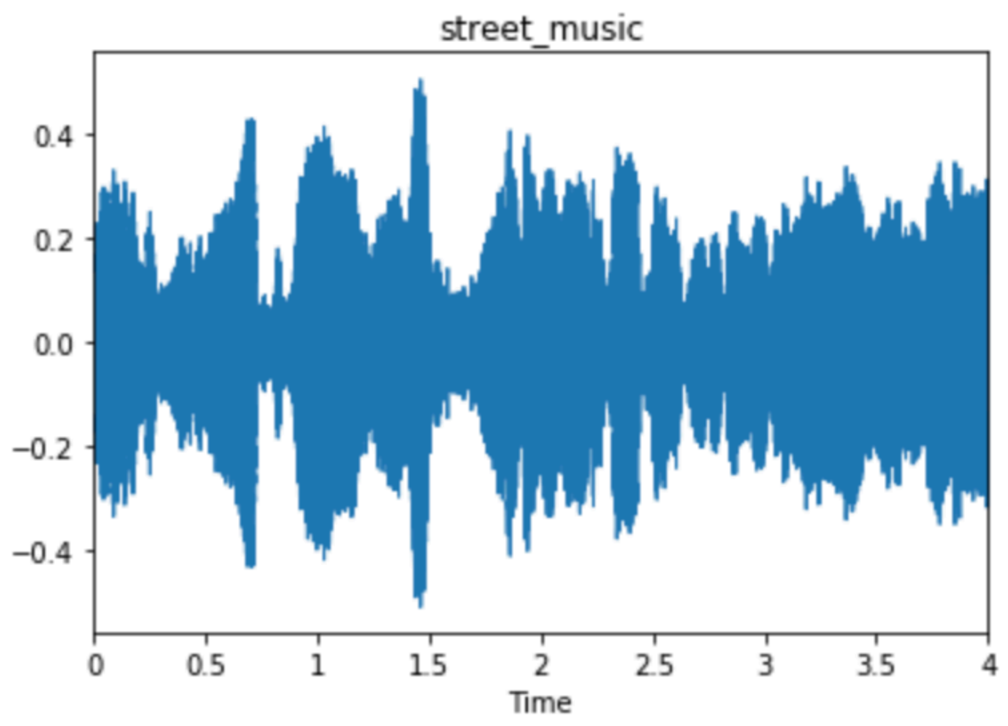

drilling
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0
0.2
0.4
0.6
0.8
1
1.2
Time


engine_idling
Time


gun_shot
Time


jackhammer
0.4
0.2
0.0
−0.2
−0.4
0
0.5
1
1.5
2
2.5
3
3.5
4
Time


siren
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0
0.5
1
1.5
2
2.5
3
3.5
4
Time



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 sold 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.

Machine Learning

Before we input our data into the model, we will split the data in a 70-30 train/test split. We will also use 5-fold cross-validation in our training.

We will model the data using a Convolutional Neural Network model. Two different CNN models of differing architecture and complexity will be fit to the data and their results compared. Due to the nature of the problem given, we will use accuracy score as our metric.

The two model architectures and their results are summarized in the table below:

	Layers	**Parameters**	**Epochs**	**Validation Accuracy**	**Test Accuracy**
Model 1	7	952,910	50	0.9416	0.9190
Model 2	12	223, 242	200	0.8988	0.8896

The first model provided a higher level of accuracy and so that is the one we will choose for our final model. The details of the model and the final confusion matrix are provided below.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv1d_1 (Conv1D)            (None, 189, 100)          600       
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 94, 100)           0         
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 90, 100)           50100     
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 45, 100)           0         
_________________________________________________________________
flatten_1 (Flatten)          (None, 4500)              0         
_________________________________________________________________
dense_1 (Dense)              (None, 200)               900200    
_________________________________________________________________
dense_2 (Dense)              (None, 10)                2010      
=================================================================
Total params: 952,910
Trainable params: 952,910
Non-trainable params: 0
_________________________________________________________________
```

```
array([[175,   1,   0,   0,   1,   1,   0,   1,   0,   1],
       [  0,  81,   0,   1,   5,   0,   0,   2,   0,   2],
       [  1,   1, 153,   7,   3,   0,   3,   0,   4,   8],
       [  3,   5,   5, 145,   2,   4,   5,   1,   6,   4],
       [  0,   0,   1,   1, 171,   0,   0,   4,   1,   2],
       [  0,   0,   2,   1,   1, 181,   0,   0,   0,   2],
       [  0,   0,   1,   1,   0,   0,  67,   0,   0,   0],
       [  1,   0,   0,   0,   1,   0,   0, 199,   0,   0],
       [  3,   0,   2,   2,   0,   0,   0,   1, 172,   2],
       [  0,   7,   4,   5,   1,   3,   0,   4,   2, 154]])
```

In examining the confusion matrix, the three classes that were most often misclassified were children playing, street music and gunshot. These were three of the classes that we noted above as having inconsistency among the wave plots and so it is perhaps not surprising that there was a level of misclassification.

Conclusions and Next Steps

The model produced encouraging accuracy results with cross-validation used. However, when the model was applied to the provided test set, accuracy score were in the low 60% range. Since the labels of the test set are unknown, it is difficult to account for these differences.

To continue the analysis, greater feature selection should be explored. Either different numerical features could surely be extracted from the sound files or a reduction of the number of features can be attempted. Perhaps some preprocessing of the audio files can be done to help reduce background noise.

Different neural network architectures can also be experimented with to try and eliminate some of this apparent overfitting.

In all though, the results obtained from our data with the known labels gives sufficient evidence that should be able to help our clients improve their security system by being able to identify dangerous vs. non-dangerous sounds.