By: Marwan Harajli

Last Updated: 11/19/18

Subject: Data Story Telling in Urban Noises Classification.

**1-Introduction:**

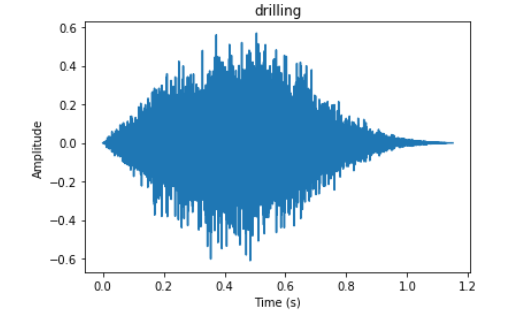
In a previous report, the identification of urban noises using supervised learning was proposed (see [here](https://docs.google.com/document/d/1Ku9QY2DAafHAPjtUHcWY-FchJzwD1pmH1bJ5vofVV2A/edit?usp=sharing)). Briefly, a database of around 8500 labeled .wav files are obtained where each file represents one of ten scenarios/classes (siren, street music, air conditioner, dog bark, jack hammer, car horn, drilling, children playing, gun shot, and engine idling). The report proposes the training of a classification algorithm using a portion of these sounds, and testing it on the remainder.

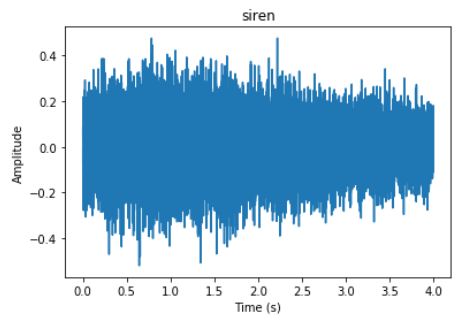
A report on data wrangling (see [here](https://docs.google.com/document/d/1tCS-aOiDy7ENlLEFcHvdyny2Wg8rgjeas0Vx5rhA-BY/edit?usp=sharing)), described the process of loading these .wav files into numpy arrays and extracting features from the arrays. These features were: Mel-frequency cepstral coefficients, mel-scaled power sepctrograms, chromograms of short-time Fourier transforms, octave based spectral constrasts, and tonnetz coefficients. The report discussed how the majority of the sounds were of 4 seconds duration, with a relatively small fraction having durations less than that. A discussion regarding the possibility of deleting these sounds with smaller durations was presented; where the strategy was found to be infeasible with some classes having the majority of their samples with durations less than 4 seconds.

This report builds on the data wrangling report, and narratively explains the transformation of sounds into usable features. The features are then analyzed, and their usability studied where the distribution of features per class is visualized for a few of them. Finally, a simple classification algorithm is described and attempted with the obtained features. The performance of the algorithm is discussed briefly.

**2-Sounds As Numpy Arrays:**

Every sound is loaded as a numpy array. Each array has a length that equals the product of duration and sampling frequency (the sampling frequency is uniform over all sounds =22050 samples/second), and every entry of the array represents an amplitude recorded at a sampled time. As mentioned, the sounds may vary in duration. We show two such loaded sounds, one representing a siren, the other a drill. Note that these sounds have different durations.





If we were to consider the amplitudes at every sampled time as a feature (to train a classificaion algorithm) we would face two problems: The first being the large number of features per sound (at most 22050 samples/s\*4s=88200 samples), the second being the varying number of features per sound (since sounds have varying durations).

**3-Features Extraction**

It is helpful to revisit the features described in the data wrangling report. In this report each of these features fi (i in [1,2,3,4,5]) is considered as a transformation that take an array **s** (the sound) of size *n*, and returns a matrix **M**iof size*mi*\**ki* (*n*,*mi*,*ki* are non-zero positive integers).

*k*i is dependent on *n* (i.e. the duration of the sound), while *m*i is dependent on the transformation itself but independent of the duration (for example how many mel bins are chosen for the mel-scaled power spectrogram). In this report, we explore the possibility of temporally averaging each **M**i to obtain a *m*i\*1 vector.

Finally, stacking these 5 vectors above each other, each sound obtains a feature (not to be confused with the features of the data wrangling report) of length:

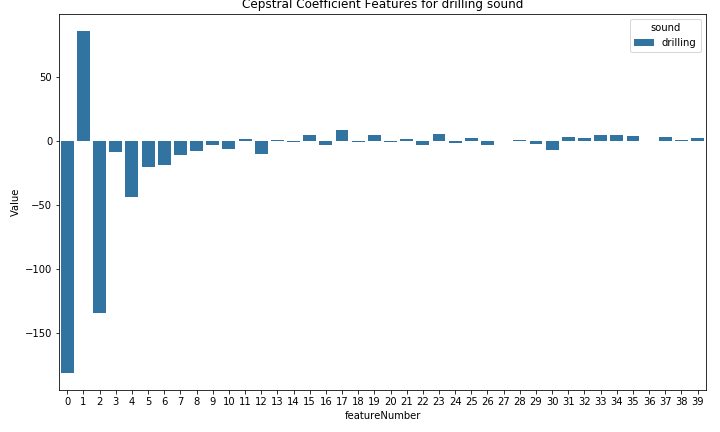
With this strategy, sounds with different durations produce features of identical lengths.

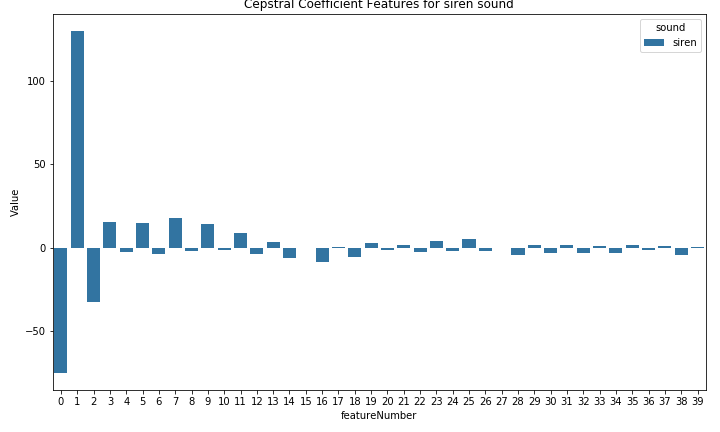
In what follows we show the 5 s for the two sounds (the drilling and the siren) shown in section 2.

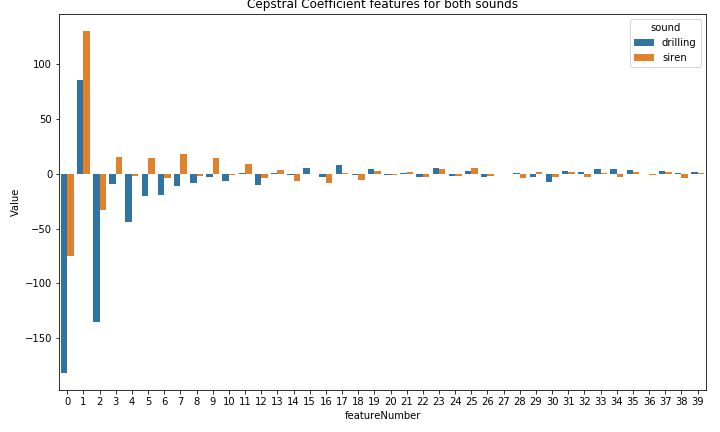
**4-Feature Visualization**:

**4.1- Mel-Frequency Cepstral Coefficients:**

This transformation produces a 40 dimensional feature as shown below for the drilling and siren sounds of section 2.

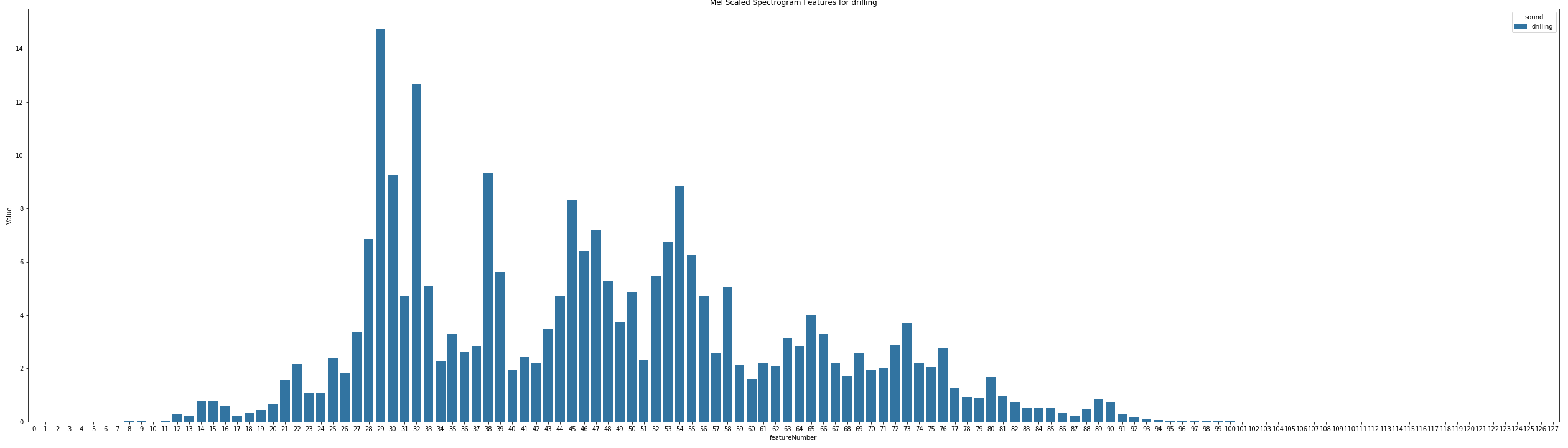


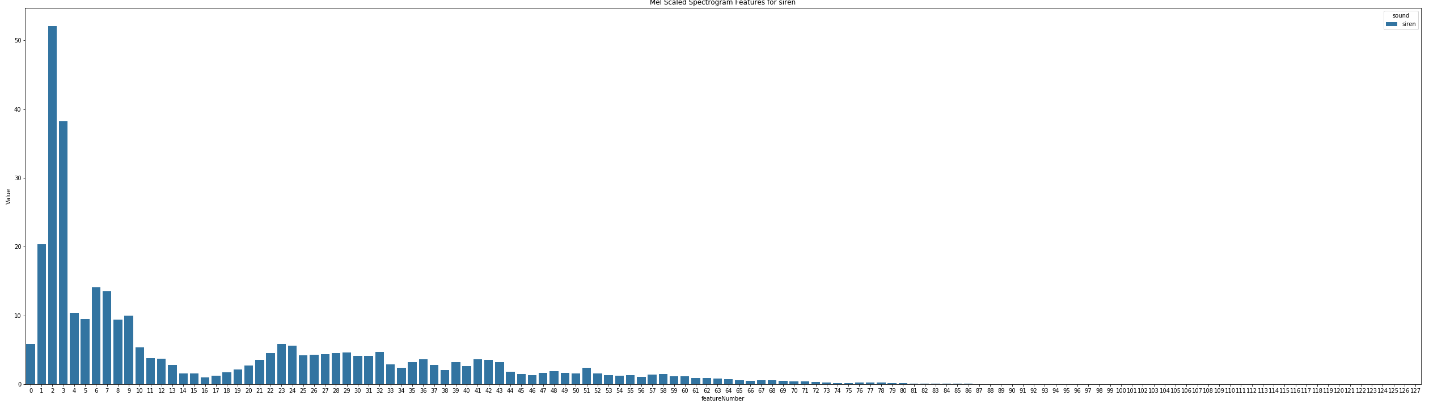


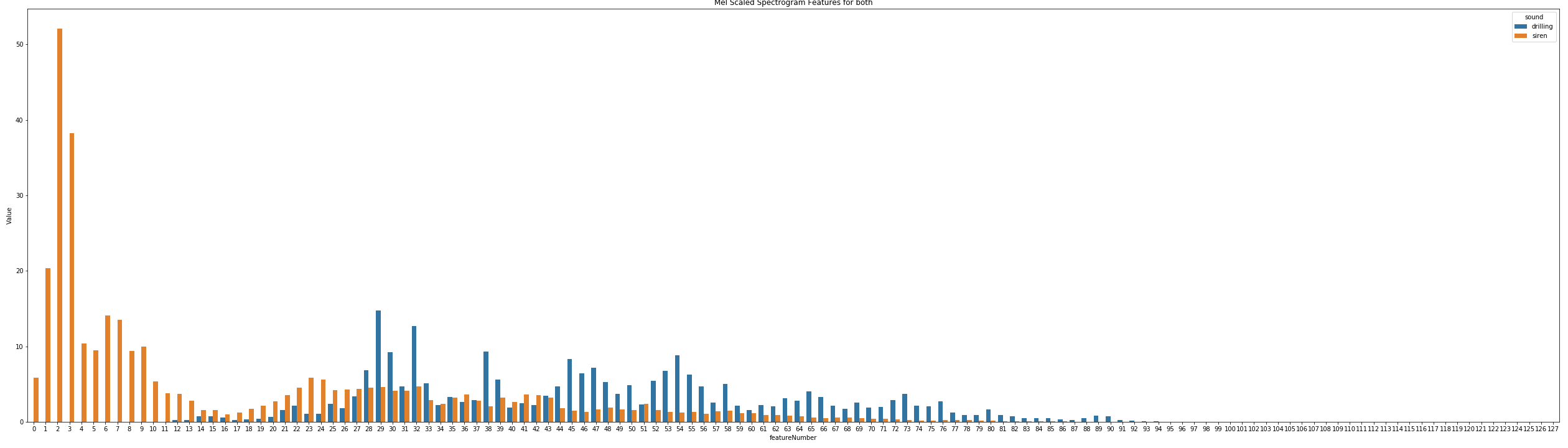


**4.2-Mel-Scaled Spectrogram:**

This transformation produces a 128 dimensional feature as shown below for the drilling and siren sounds of section 2.

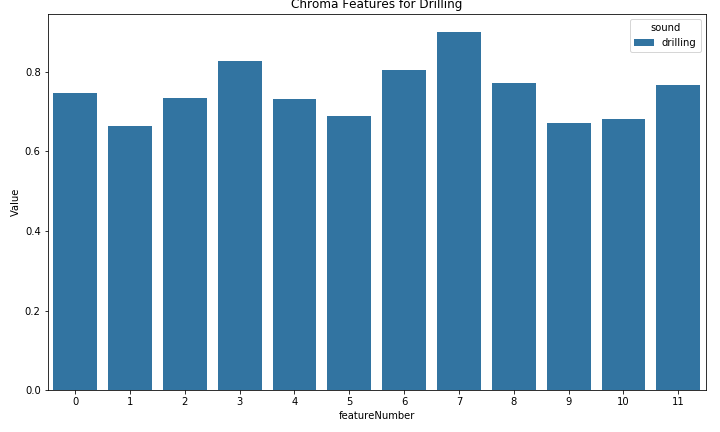


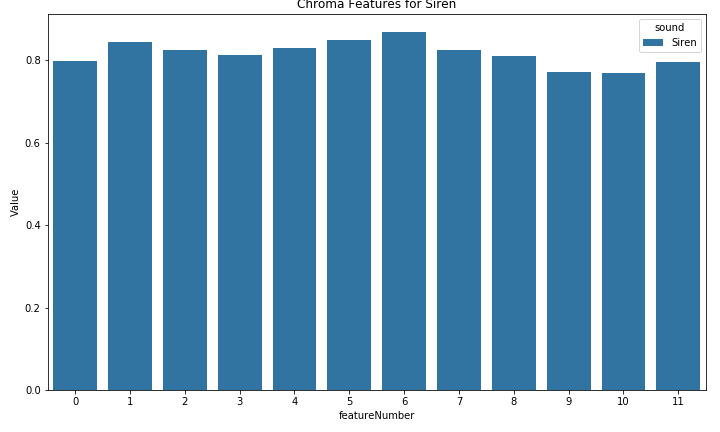


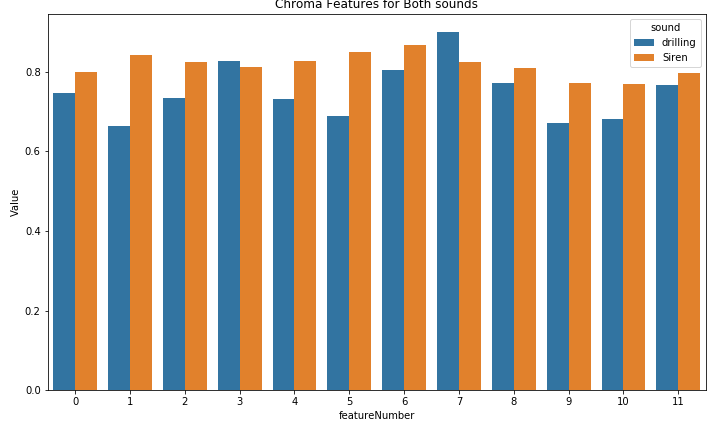


**4.3-Chroma of a short-time Fourier Transform:**

This transformation produces a 12 dimensional feature as shown below for the drilling and siren sounds of section 2.

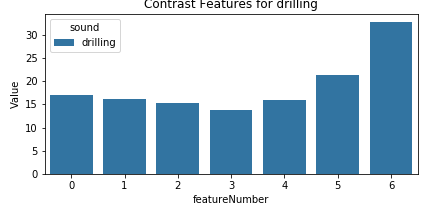


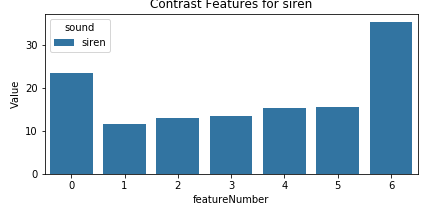


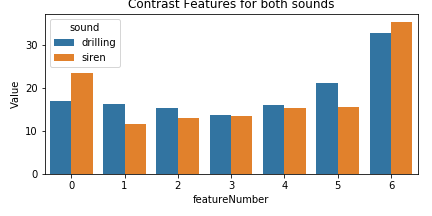


**4.4-Octave Based Contrast:**

This transformation produces a 7 dimensional feature as shown below for the drilling and siren sounds of section 2.

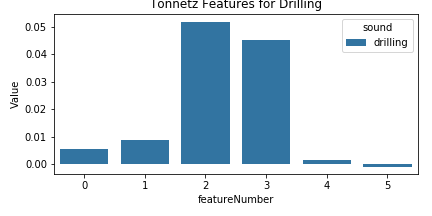


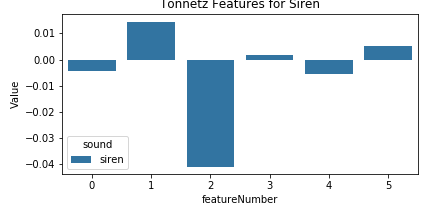


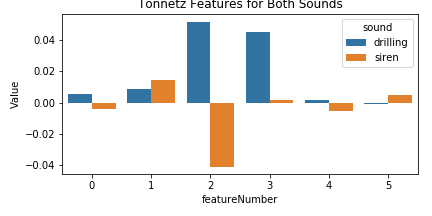


**4.5-Tonnetz:**

This transformation produces a 6 dimensional feature as shown below for the drilling and siren sounds of section 2.

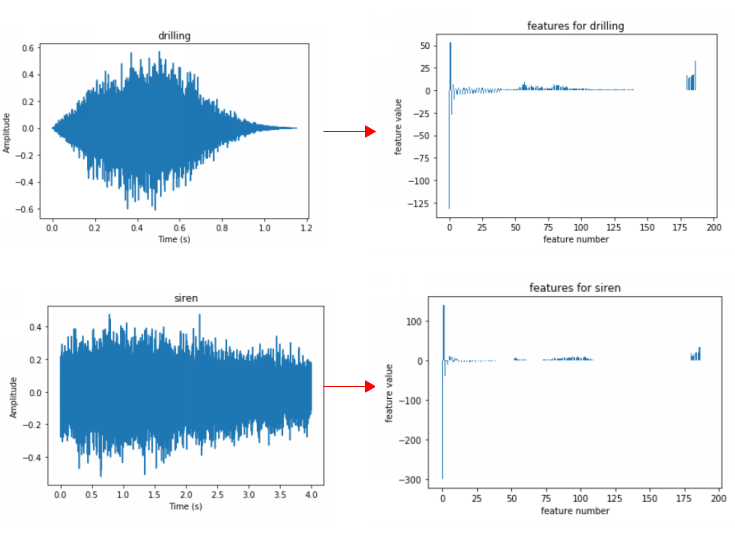




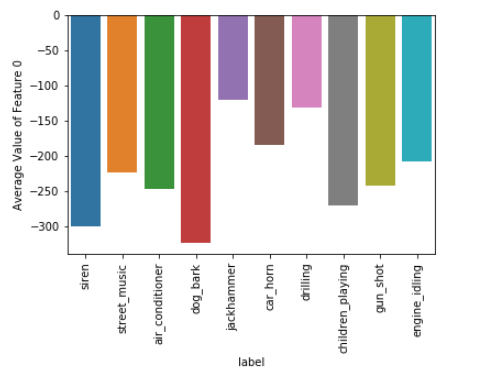


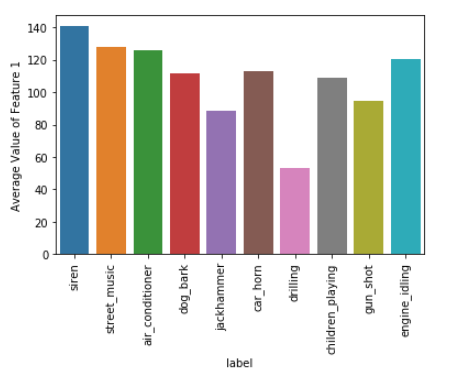
**5- Analysis of Extracted Features:**

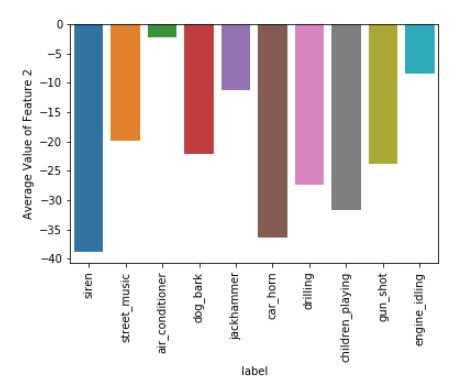
With the above, each sounds is transformed from a numpy array of varying size into a 193 dimensional feature vector. The siren and drilling sounds used in section 2 are shown below with their respective extracted features.



A natural question to ask is if the features vary in distribution from class to another. For example, if a feature is identically distributed over all classes then the feature does not provide much information about the class of the sound it is extracted from. As a start, we calculate the mean of every feature for every class. With that obtained, it could be informative to check the mean of a given feature for varying classes. We show the means of features 0,1, and 2 (the first 3 MFCC coefficients) for different classes below.

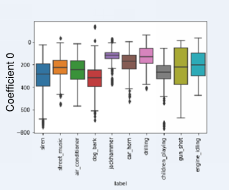


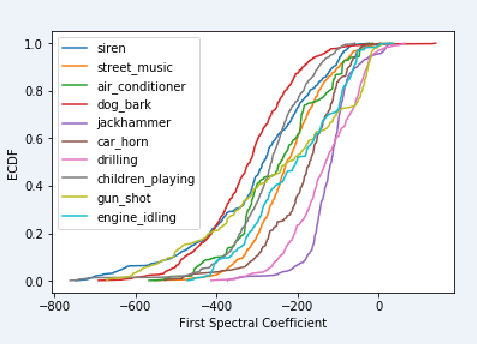




From the above three plots, we can see that for example features 0 and 1 and not great in differentiating between street music and air conditioner sounds, but feature 2 is.

Other visualizations we can use to see variability in features between classes are box plots and ECDFs as shown below for feature 0 (the first spectral coefficient).

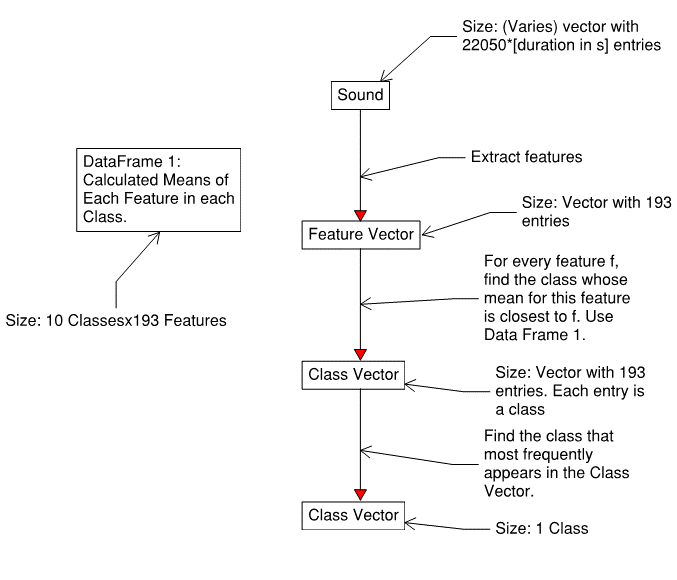




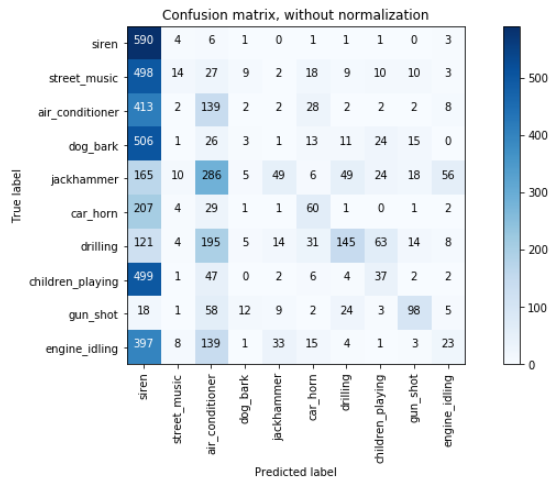
**7- A Simple classification Algorithm and its Evaluation:**

In what follows we classify sounds based on a simple classification algorithm. The algorithm itself is not sophisticated and the general methodology is not scientifically correct (for example, we are obtaining means of all features over the different classes and then using these means to classify the same data). However, the process can shed some light about the usefulness of the extracted features.

The algorithm is described in the flow chart shown below:



A confusion matrix is generated to check the performance of the algorithm above.



As can be seen, the algorithm generally misclassifies sounds as siren sounds.

**8- Conclusions:**

In this report the extraction of features is analyzed and visualized for a few selected sounds. The distribution of these features is assessed for varying classes and graphics are obtained to show these distributions.

The graphics and code for this report can be found in [this](https://github.com/harajlim/Urban_Noises/blob/master/Data%20Story%20Telling%20111918.ipynb) github repo.

In what follows, more involved algorithms are explored and their performance assessed in a more rigorous setting.