**Statement**

This is a classification problem and is meant to let me have hands on exposure to audio processing in the usual classification scenario. Compared to other bodily features, voice is dynamic and complex. And voice classification has found useful applications in classifying speakers' gender, mother tongue or ethnicity (accent), emotion states, identity verification, verbal command control, and so forth. Furthermore voice classification potentially can apply to interactive-voice-response system for detecting the moods and tones of customers, thereby guessing if the calls are of complaints or complement, for example. As a consequence, digging into audio data can be beneficial.

Notes: the data came from <https://datahack.analyticsvidhya.com/contest/practice-problem-urban-sound-classification/#ProblemStatement>. It’s a popular website where there’re lots of competitions going on for data geeks to learn and explore.

**Introduction to datasets**

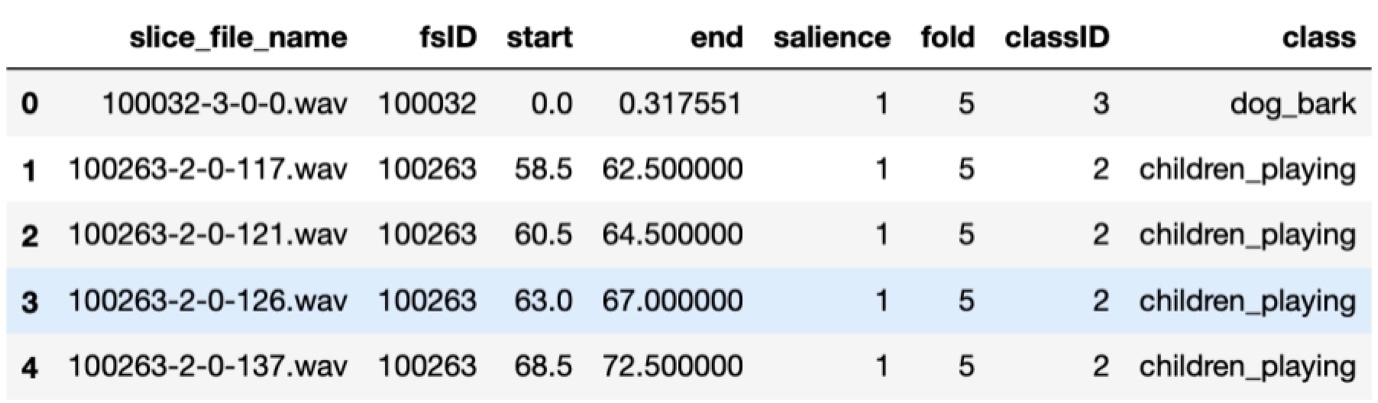
This dataset contains 8732 labeled sound excerpts (<=4s) of urban sounds from 10 classes:

air\_conditioner, car\_horn, children\_playing, dog\_bark, drilling, enginge\_idling, gun\_shot,

jackhammer, siren, and street\_music. Besides, the dataset has already been split into train and test datasets. The training dataset includes 5435 samples, while the testing dataset includes 3297 samples. And there’s no missing value in the dataset. The classes are drawn from the urban sound taxonomy. For a detailed description of the dataset and how it was compiled please refer to online explanations. All excerpts are taken from field recordings uploaded to [www.freesound.org](http://www.freesound.org).

In addition to the sound excerpts, a CSV file containing metadata about each excerpt is also provided.

**1. Details on datasets**



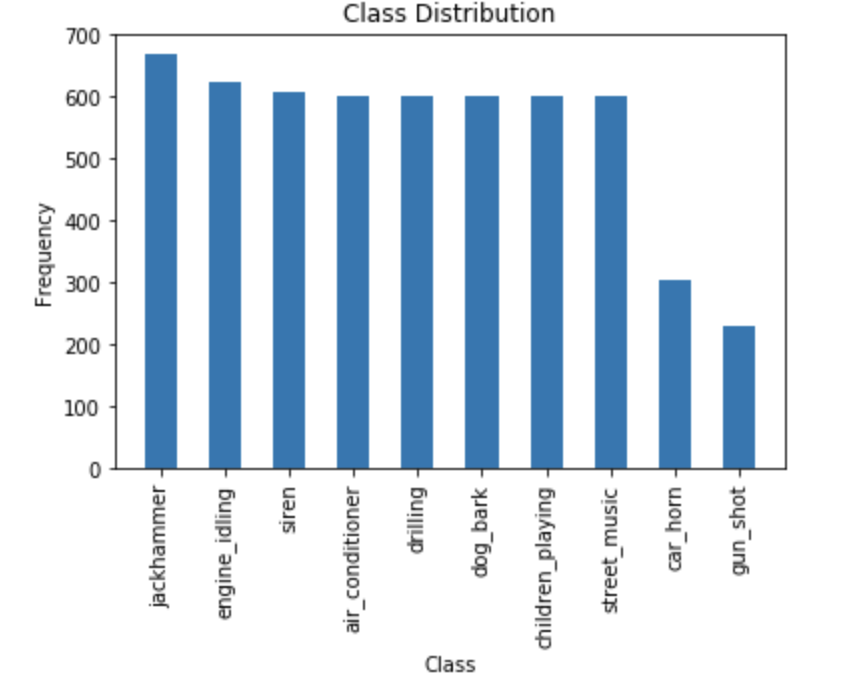
Above is a grasp ofCSV file, below is the details of each cloumns.

--Column Names

* slice\_file\_name: The name of the audio file. The name takes the following format: [fsID]-[classID]-[occurrenceID]-[sliceID].wav, where: [fsID] = the Freesound ID of the recording from which this excerpt (slice) is taken [classID] = a numeric identifier of the sound class (see description of classID below for further details) [occurrenceID] = a numeric identifier to distinguish different occurrences of the sound within the original recording [sliceID] = a numeric identifier to distinguish different slices taken from the same occurrence
* fsID: The Freesound ID of the recording from which this excerpt (slice) is taken
* start The start time of the slice in the original Freesound recording
* end: The end time of slice in the original Freesound recording
* salience: A (subjective) salience rating of the sound. 1 = foreground, 2 = background.
* fold: The fold number (1-10) to which this file has been allocated.
* classID: A numeric identifier of the sound class: 0 = air\_conditioner 1 = car\_horn 2 = children\_playing 3 = dog\_bark 4 = drilling 5 = engine\_idling 6 = gun\_shot 7 = jackhammer 8 = siren 9 = street\_music
* class: The class name: air\_conditioner, car\_horn, children\_playing, dog\_bark, drilling, engine\_idling, gun\_shot, jackhammer, siren, street\_music.

This information is quite important, because we need to use it as a specific form to extract our data.

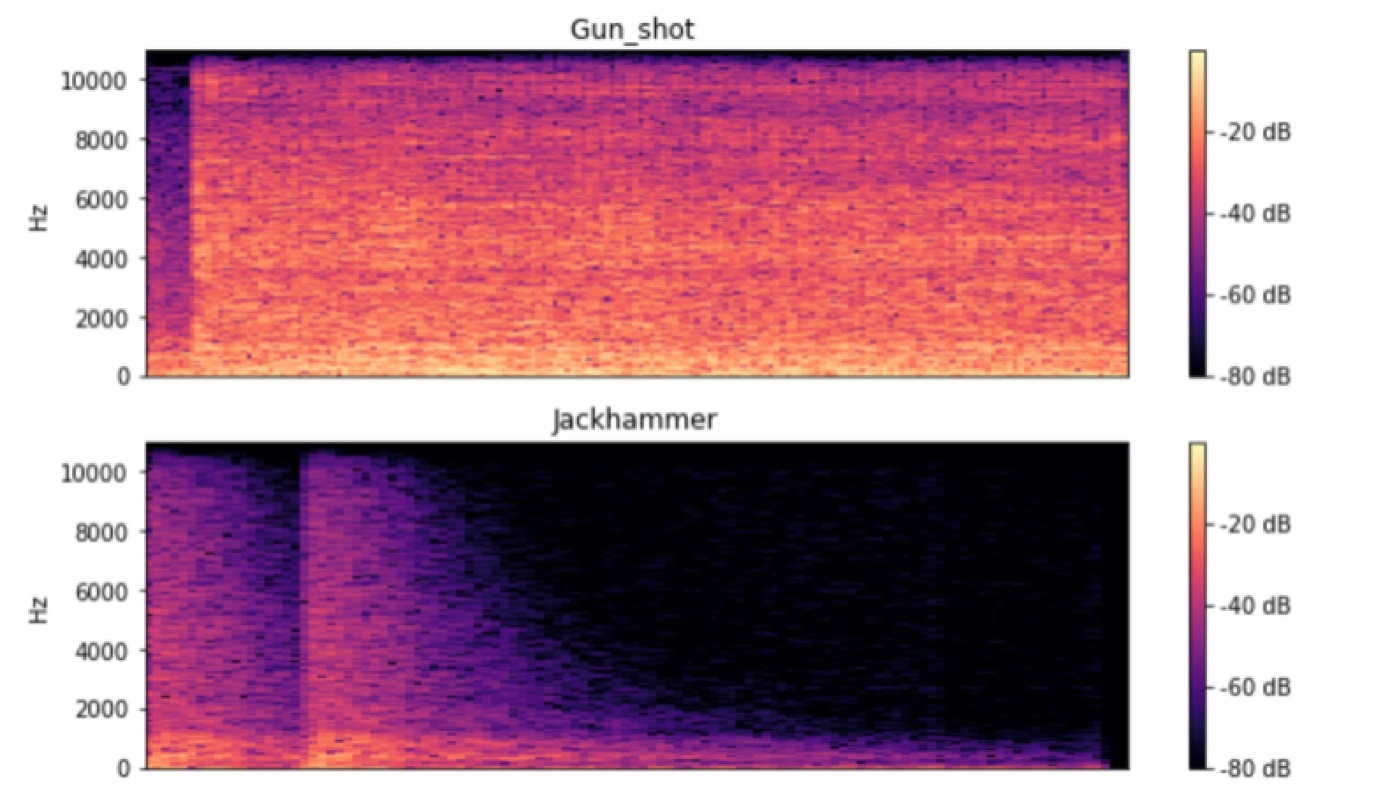
Below is the distribution of each class.



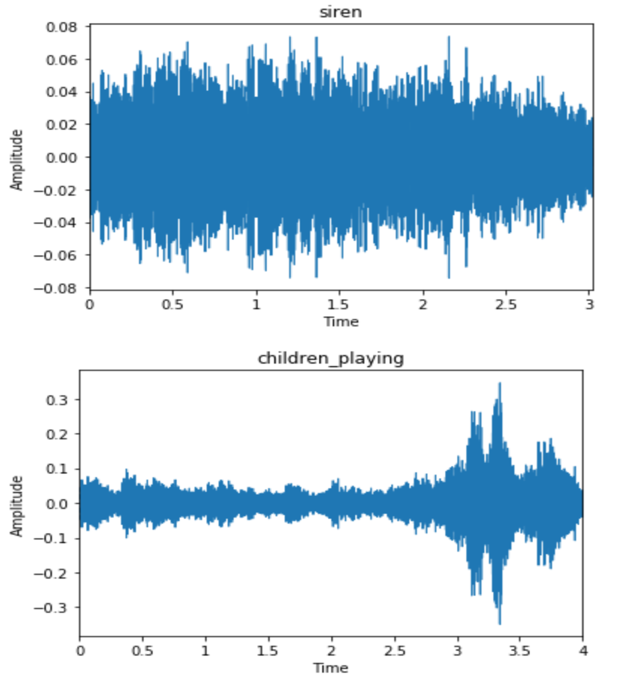
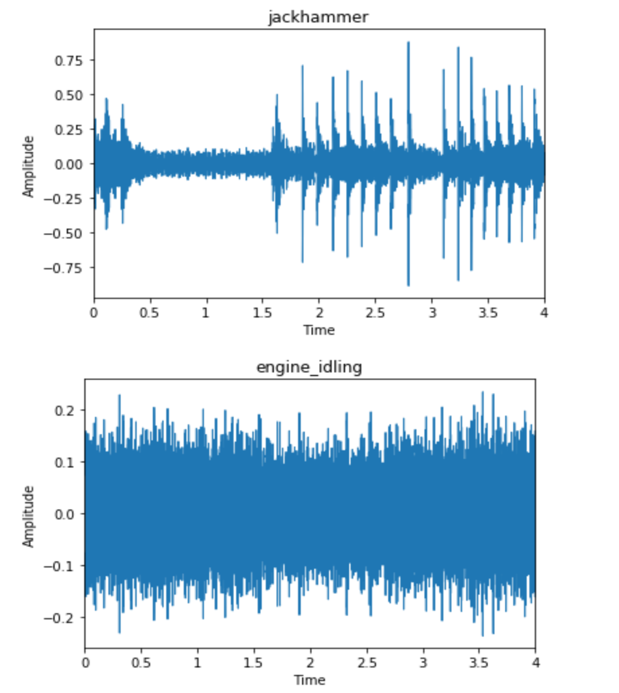
We can see that the car\_horn and gun\_shot class are much less that others, while the number of other classes are quite similar.

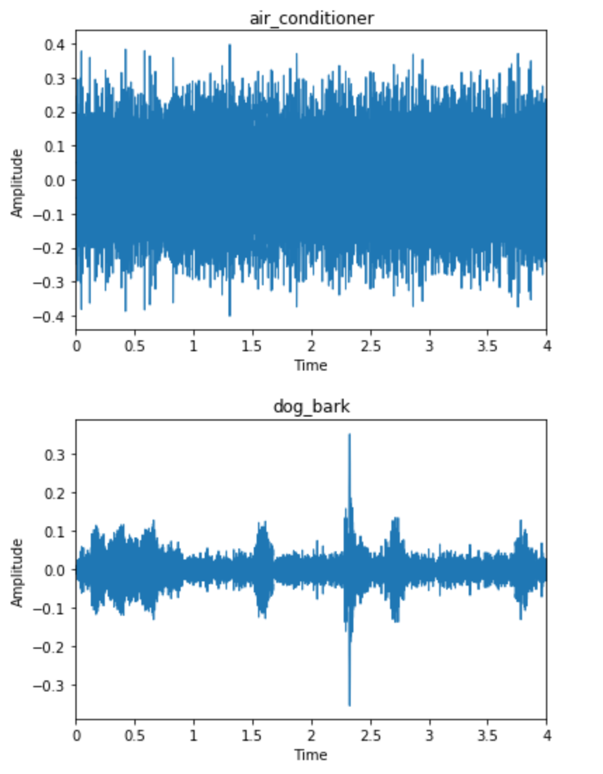
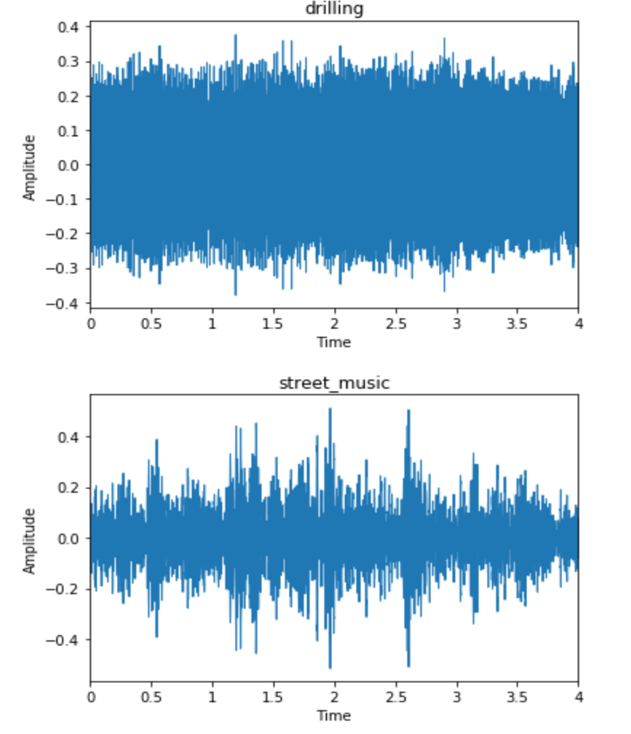
**2. Visualization**

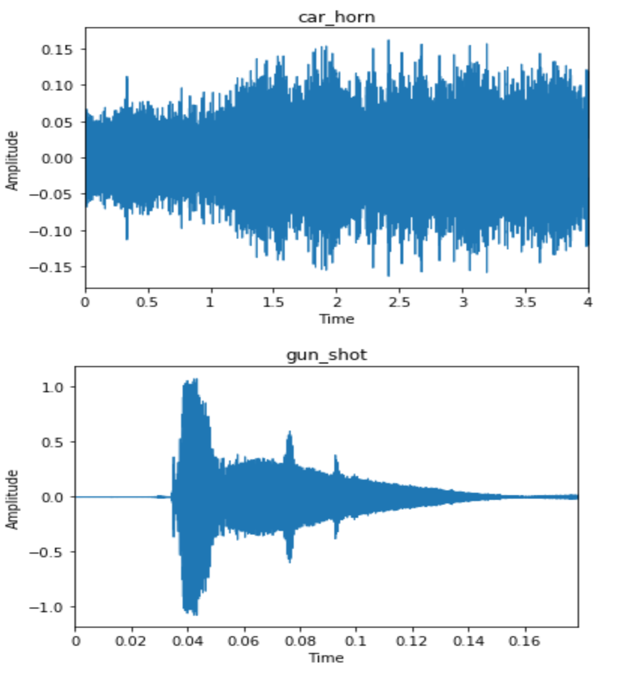
I also usedLibrosa package to visualize different classes. We can see that, different class has different waveforms.



To be more straight forward, I plot the wave plot of each class, all of them seems quite different.







**Feature Extraction and Database Building**

1. I have used Librosa to extract features. For features, I tried both Mel-spectrogram and MFCC.

2.After reshaping and cleaning the data, I split the training data into train and validate data. The splitting was based on the training data only.

3. Labels were converted to Categorically Encoded Data.

Note : Extracting features may take upto 45 minutes depending on your hardware since it has to extract spectogram data for 8732 audio files. As a result, I did this step on remote server which took me 10 minutes.

**Model Selection**

## 1. I planned to employ both Multilayer Perceptron Model (MLP) and Convolutional Neural Network (CNN)