



Distinguish Sound With Machine Learning.

Urban Sound Classification

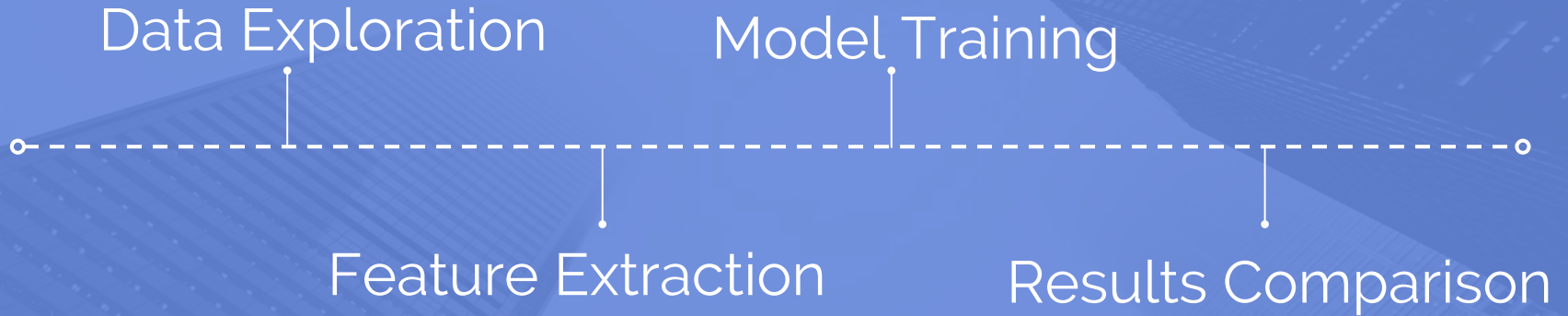
Group 4:
Chen, Tiantian
Li, Yijia
Lin, Han
Shi, Qian
Zhou, Xiaoyu

OVERVIEW

- **Background:** exposed to different sounds every day
- **Goal:** to classify sounds automatically
- **Challenge:** compared to image, feature extraction for sound is not quite straightforward



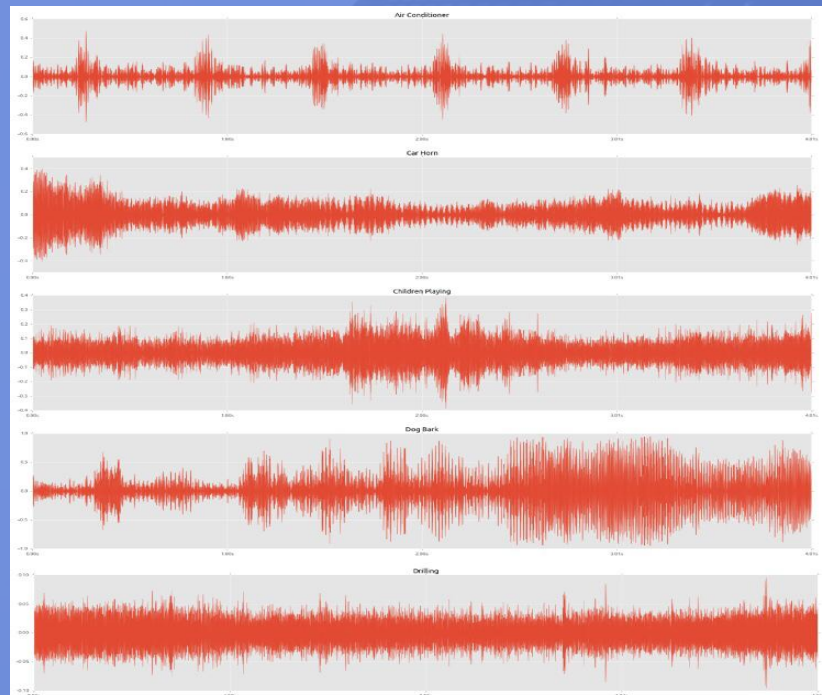
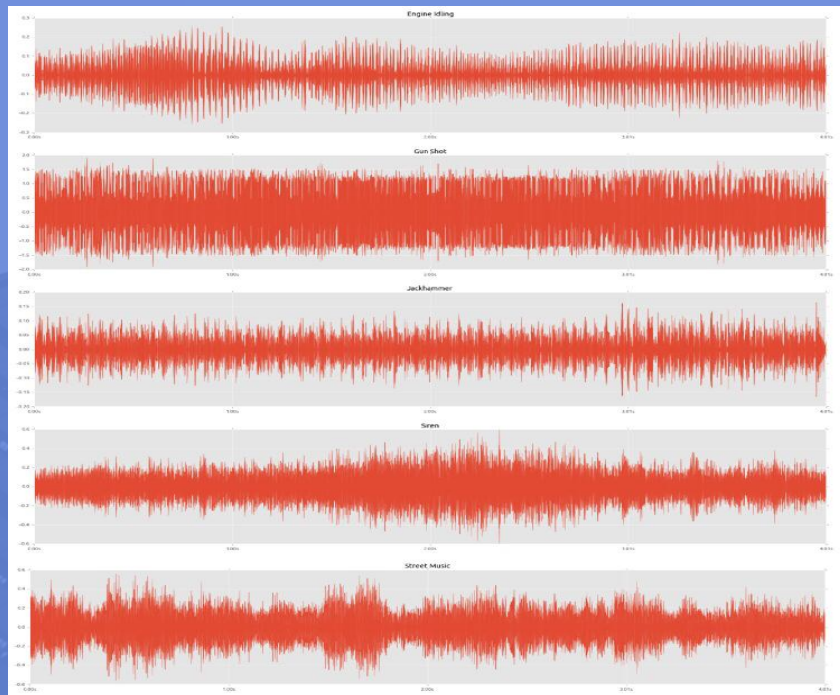
OUR PROCESS FOLLOWS:



Data

1. **Source:** Urban Sound Dataset
2. **Size:** 1302 labeled sound recordings
3. **Format:** each recording is a wav file
4. **Content:** 10 classes of sounds (air_conditioner, car_horn, children_playing, dog_bark, drilling, engine_idling, gun_shot, jackhammer, siren, and street_music)
5. **Usage:** 30% of the data is regarded as test data

Waveplot



Feature Extraction

Tool: Librosa library in Python

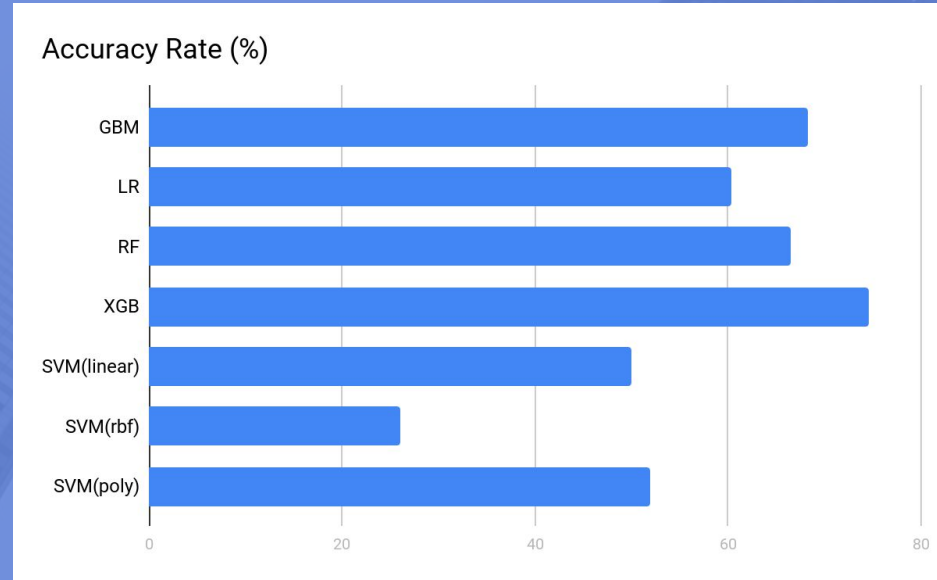
Methods:1

- **melspectro gram:** Compute a Mel-scaled power
- **mfcc:** Mel-frequency cepstral coefficients
- **chorma-stft:** chromagram from a waveform/power spectrogram
- **spectral_contrast:** Compute spectral contrast, using method
- **tonnetz:** Computes the tonal centroid features (tonnetz)

Output: Matrix of 1102 rows, 193 columns

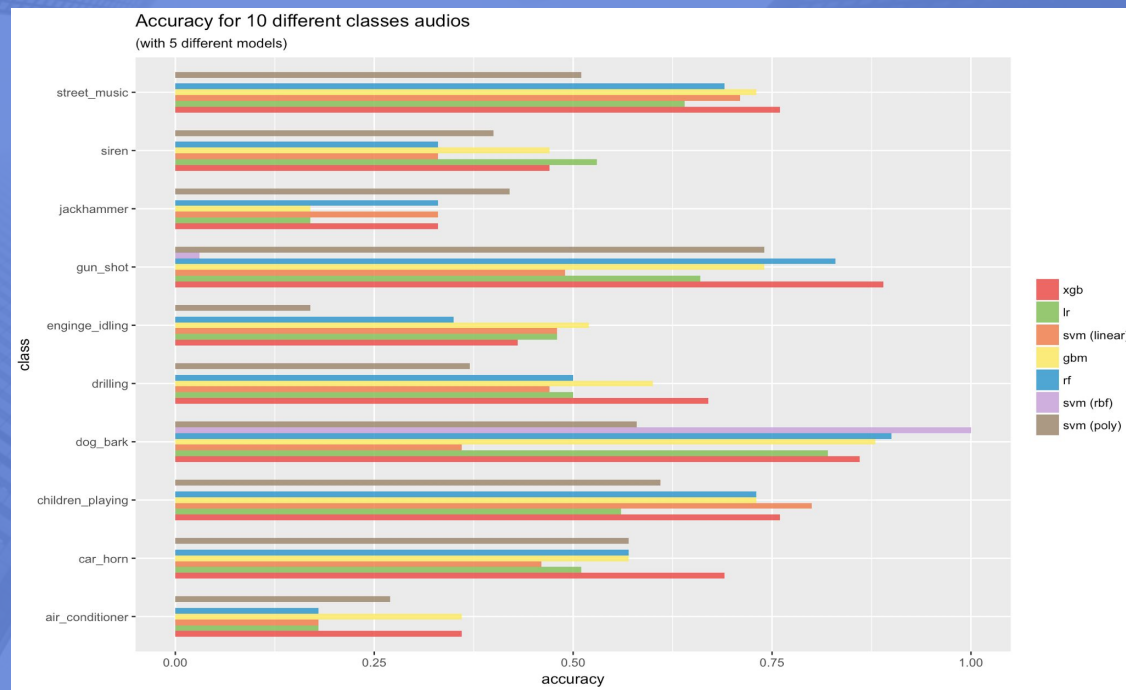
Model Comparison

Accuracy Rate:



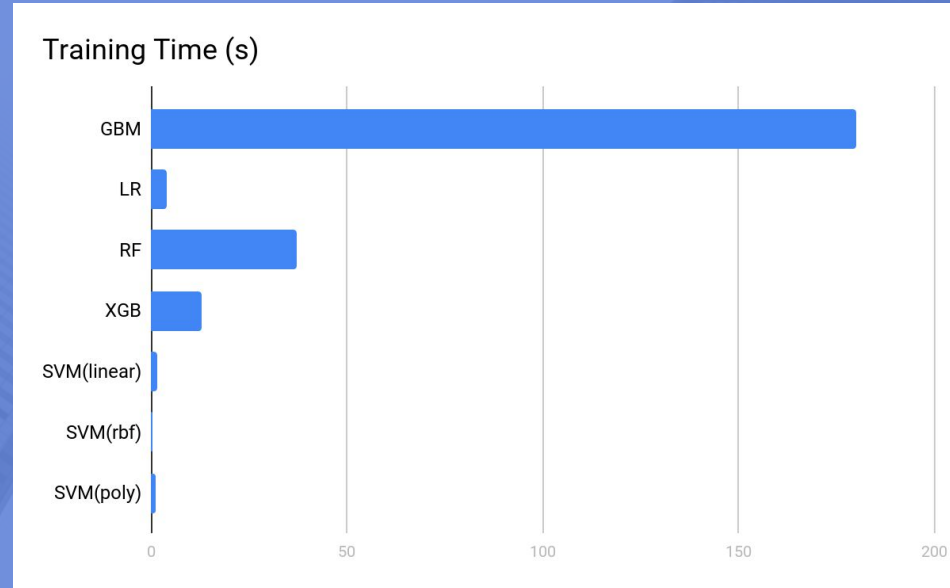
Model Comparison

Accuracy for
Each Class:



Model Comparison

Length of
Training Time:



Model Comparison

| Model | Accuracy Rate (%) | Training Time(s) |
|-------------|-------------------|------------------|
| GBM | 68.28 | 1200 |
| LR | 60.42 | 3.96 |
| RF | 66.47 | 37.09 |
| XGB | 74.61 | 13 |
| SVM(linear) | 50 | 1.62 |
| SVM(rbf) | 26 | 0.43 |
| SVM(poly) | 52 | 1 |

improvements

- Use a larger dataset on the official website which contains more than 8000 audio records with these 10 different classes. Intuitively, more data available, more accurate for model performs.
- Set up a more complex CNN with at least 10 layers, and with some other techniques, such as adding zero-padding or dropout layer to classify those audios.
- Try to use different combinations within those five kinds of features we extracted. We have assumed that perhaps some kinds of the features are much more outstanding than others. So why not try to ignore those "unuseful" features, at some points, reduce dimensions, and train the model on the subset of the features. The result could be exciting, or not.
- Try to extract other kinds of features.



THANKS

Any questions?

You can find our projects at
<https://github.com/TZstatsADS/fall2017-project5-grp4>