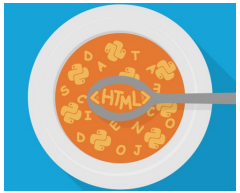
Classifying Samples by Instrument

Why Intelligent Pre-processing Matters

Data Sources



amazon web services

Amazon S3

MySQL

Data scraped using BeautifulSoup (a Python web-scraping library) from the following sources:

- University of Iowa Electronic Music Studios Musical Instrument Sample Database

  http://theremin.music.uiowa.edu/MIS.html

  University of Iowa Electronic Music Studios

- UK Philharmonia Orchestra Sound Samples
  - https://www.philharmonia.co.uk/explore/sound_samples

philharmonia orchestra
Esa-Pekka Salonen
Principal Conductor & Artistic Advisor

Samples loaded in AWS S3, with sample metadata stored in an AWS MySQL database.

Problem

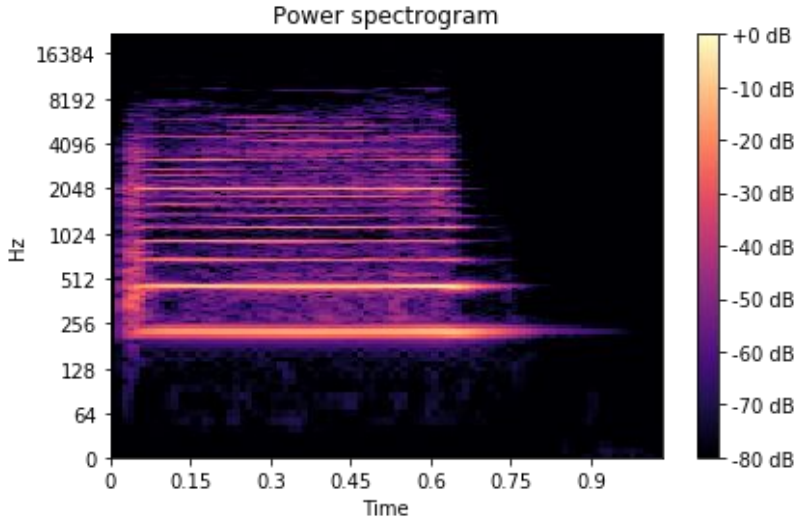
Task: Given a short audio sample of an instrument playing, identify which instrument it is

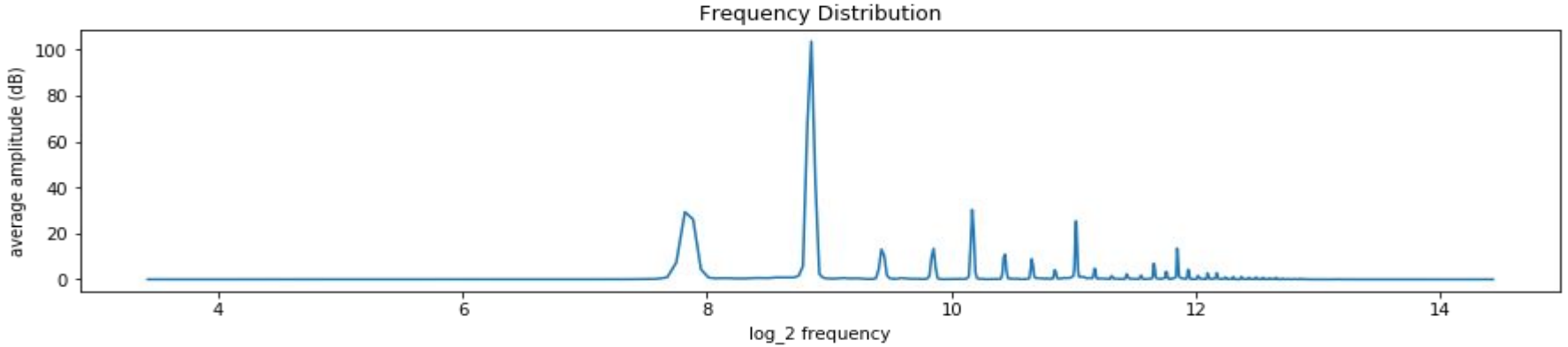
Restrictions: only detect pitched instruments commonly present in a full orchestra

Number of Classes: 24 different instruments

Baseline Model

- Spectral content measured by short-time Fourier transform (stft) averaged over all frames in the sample.
- Random Forest model with quick gridsearch gives ~15% accuracy rating. But is this good?





```
df3['Instrument'].value_counts()
```

```
Marimba           364
Vibraphone        209
Bass              208
Viola             200
Cello             195
Violin            182
Xylophone         175
bells              82
Flute              77
Trumpet            71
AltoSax            64
SopSax             64
BbClarinet         46
BassClarinet       46
Horn               44
Bassoon            40
EbClarinet         39
BassFlute          38
Tuba               37
AltoFlute          36
Oboe               35
TenorTrombone      33
BassTrombone       27
Crotale            24
Name: Instrument, dtype: int64
```

It looks like 15% of our samples are from the marimba, due to many different combinations of mallets and technique being sampled.

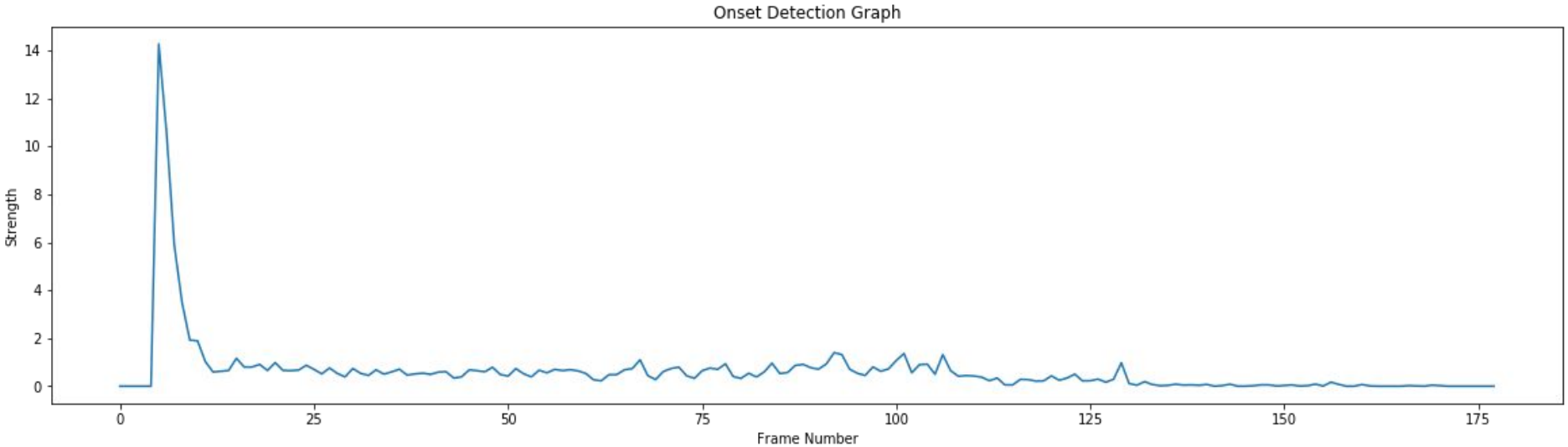
We can check that our baseline model simply guesses "Marimba" no matter the input!

```
100*df3['Instrument'].value_counts(normalize=True)
```

```
Marimba           15.582192
Vibraphone         8.946918
Bass               8.904110
Viola              8.561644
Cello              8.347603
Violin             7.791096
Xylophone          7.491438
bells              3.510274
Flute              3.296233
Trumpet            3.039384
AltoSax            2.739726
SopSax             2.739726
BbClarinet         1.969178
BassClarinet       1.969178
Horn               1.883562
Bassoon            1.712329
EbClarinet         1.669521
BassFlute          1.626712
Tuba               1.583904
AltoFlute          1.541096
Oboe               1.498288
TenorTrombone      1.412671
BassTrombone       1.155822
Crotale            1.027397
Name: Instrument, dtype: float64
```

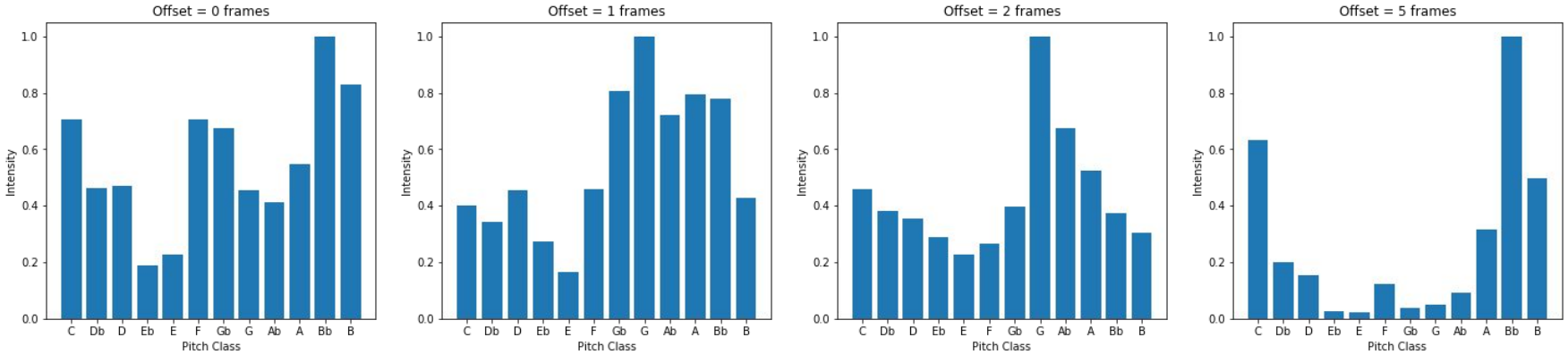
Improving the Baseline: Preprocessing with librosa

- Onset detection to isolate attack within audio sample



Preprocessing with librosa

- Onset detection to isolate attack within audio sample
- Chromagrams at centisecond-scale frames to give a low-dimensional picture of the shape of decay of harmonic content after the attack.



Preprocessing with librosa

- Onset detection to isolate attack within audio sample
- Chromagrams at centisecond-scale frames to give a low-dimensional picture of the shape of decay of harmonic content after the attack.
- Additional features: centroid, contrast, flatness, rolloff, and Tonnetz dimensions for spectrum near attack.

Altogether, reduces dimension of feature space from ~100,000 to 112.

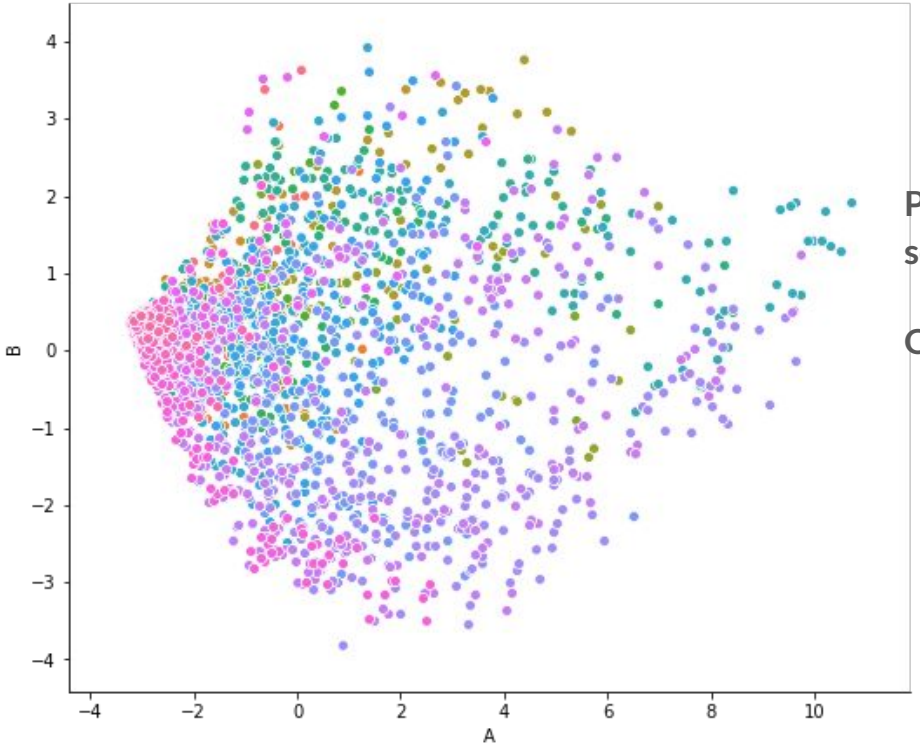
A peak at the metadata

N ~ 2400 lossless audio samples, a few seconds long each, tagged with metadata.

Expression and note were not used in training, only for troubleshooting.

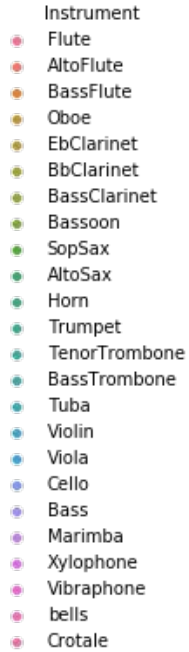
sample_id	instrument_name	note	expression	source	file_extension
2039	Vibraphone	E5	sustain	Iowa2012	aif
1611	Marimba	Db4	cord	Iowa2012	aif
1550	Marimba	Gb5	roll	Iowa2012	aif
46	Flute	E4	nonvib	Iowa2012	aif
2215	Vibraphone	Eb6	bow	Iowa2012	aif

Further Dimension Reduction Using PCA?



PCA with 5, 10, or 20 variables and standard scaling turns out not to help.

Our original 112 variables are quite informative.



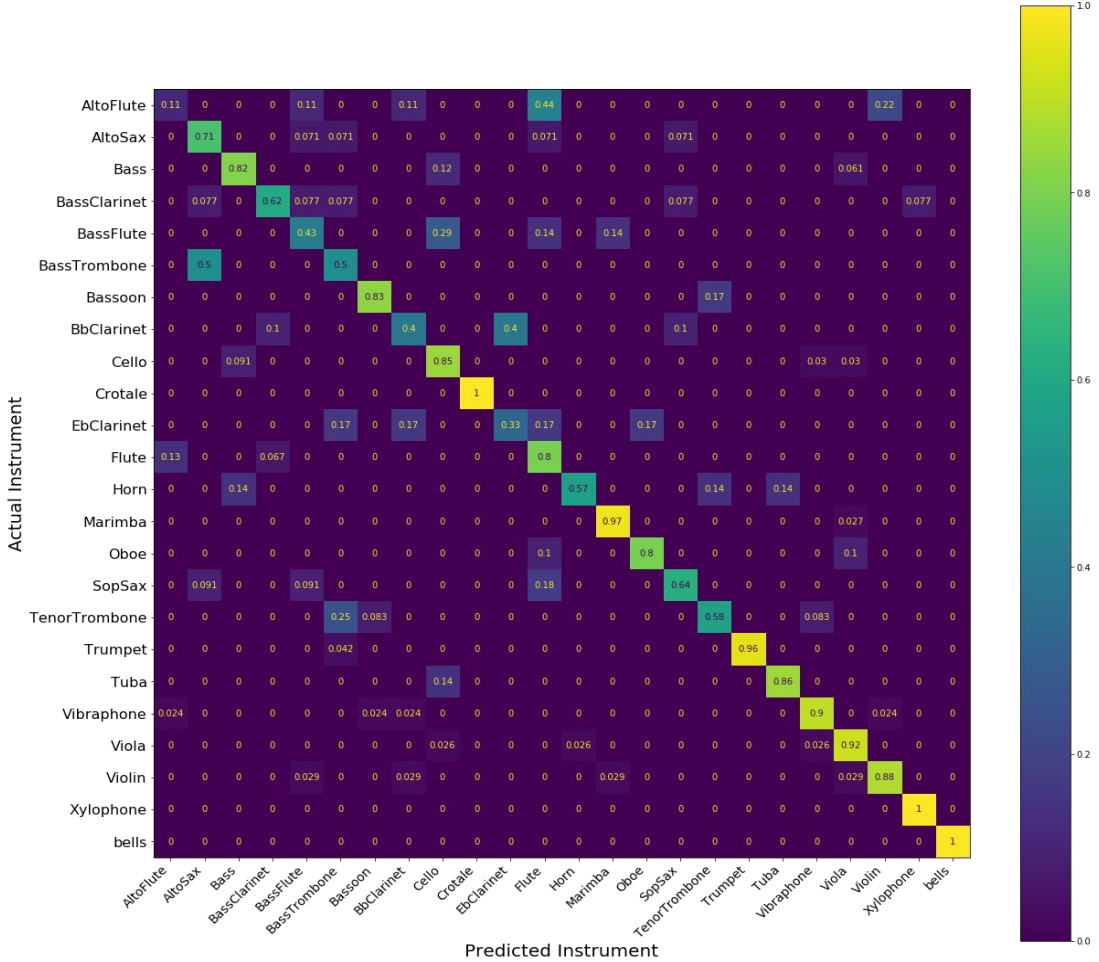
XGBoost

Parameters:

- # Estimators = 500
- Learning Rate = 0.1
- Max Depth = 3

Accuracy Score: 83%

This is quite good for a 24-class problem with roughly equal classes! This can be seen through the normalized confusion matrix.



Conclusions and Further Questions

Intelligent dimension reduction can eliminate the need for Principal Component Analysis (PCA).

Percussion instruments tend to have more distinctive sounds (more easily isolated by XGBoost).

Clarinets are hard to isolate with XGBoost.

Can ensemble methods be used to improve classification for instruments XGBoost confuses?