Machine Learning and Pattern Classification

JOHANNES KEPLER
UNIVERSITÄT LINZ

Labeling Functions & Audio Features:
Team Observe

Johannes Grafinger, Jonas Gantar, Leonhard Markus Spanring, Reinhard Josef Pötscher

JOHANNES KEPLER
UNIVERSITÄT LINZ
Altenberger Straße 69
4040 Linz, Österreich
jku.at

Labeling Function

Assess how accurately the applied labeling functions capture the intended classes.

Class	Keyword 1	Count	Keyword 2	Count	Keyword 3	Count	Total per Label
Alarm	alarm	359	beeps	110	clock	78	547
Beep/Bleep	beep	251	beeps	231	beeping	193	675
Car	car	741	engine	333	driving	179	1253
Speech	speaking	1166	talking	602	man	608	2376
Hammer	hammer	259	hammering	147	metallic	98	504
Siren	siren	351	police	80	ambulance	61	492
Bell	bell	655	rings	442	ringing	231	1328
Shout	shouting	187	loudly	208	screams	130	525

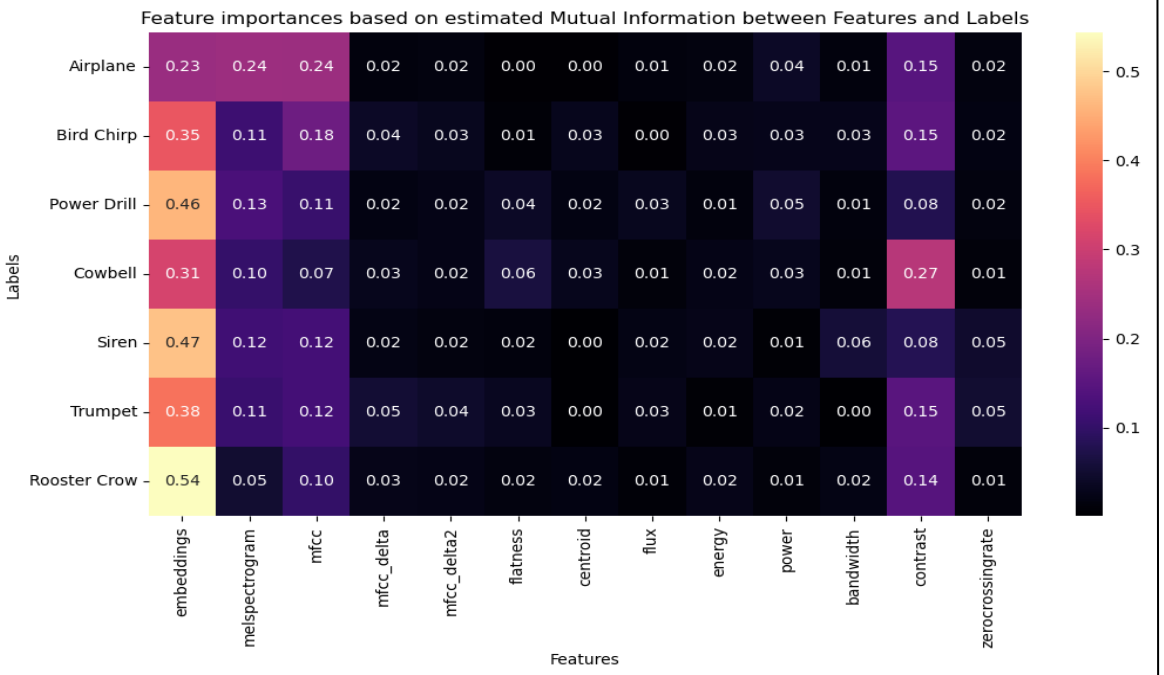
-> Class labels seem to match the textual annotations quite well

For our further analysis we used **single frames of the spectrogram + 2 frame context window** on both sides. We focus on a **subset of 7 randomly chosen classes**.

Which audio features appear most useful for distinguishing between the classes of interest?

Mutual Information:

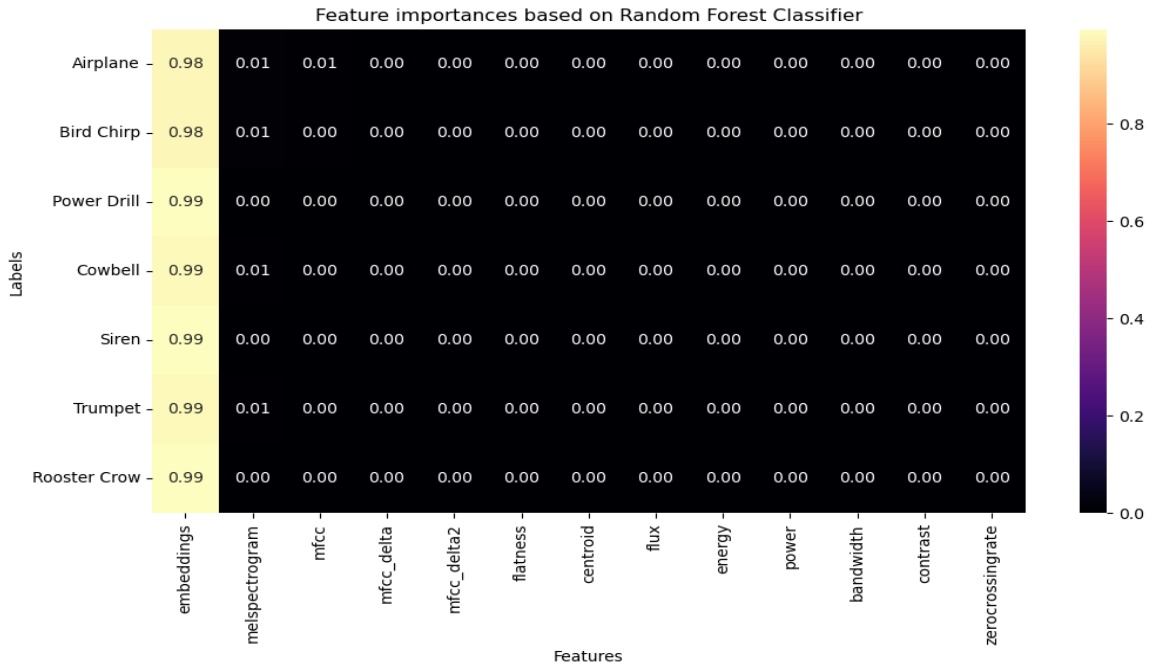
- Nonlinear relationships, no combinatorial effects



Seemingly useful features: embeddings, melspectrogram, mfcc, contrast

RF Feature importances

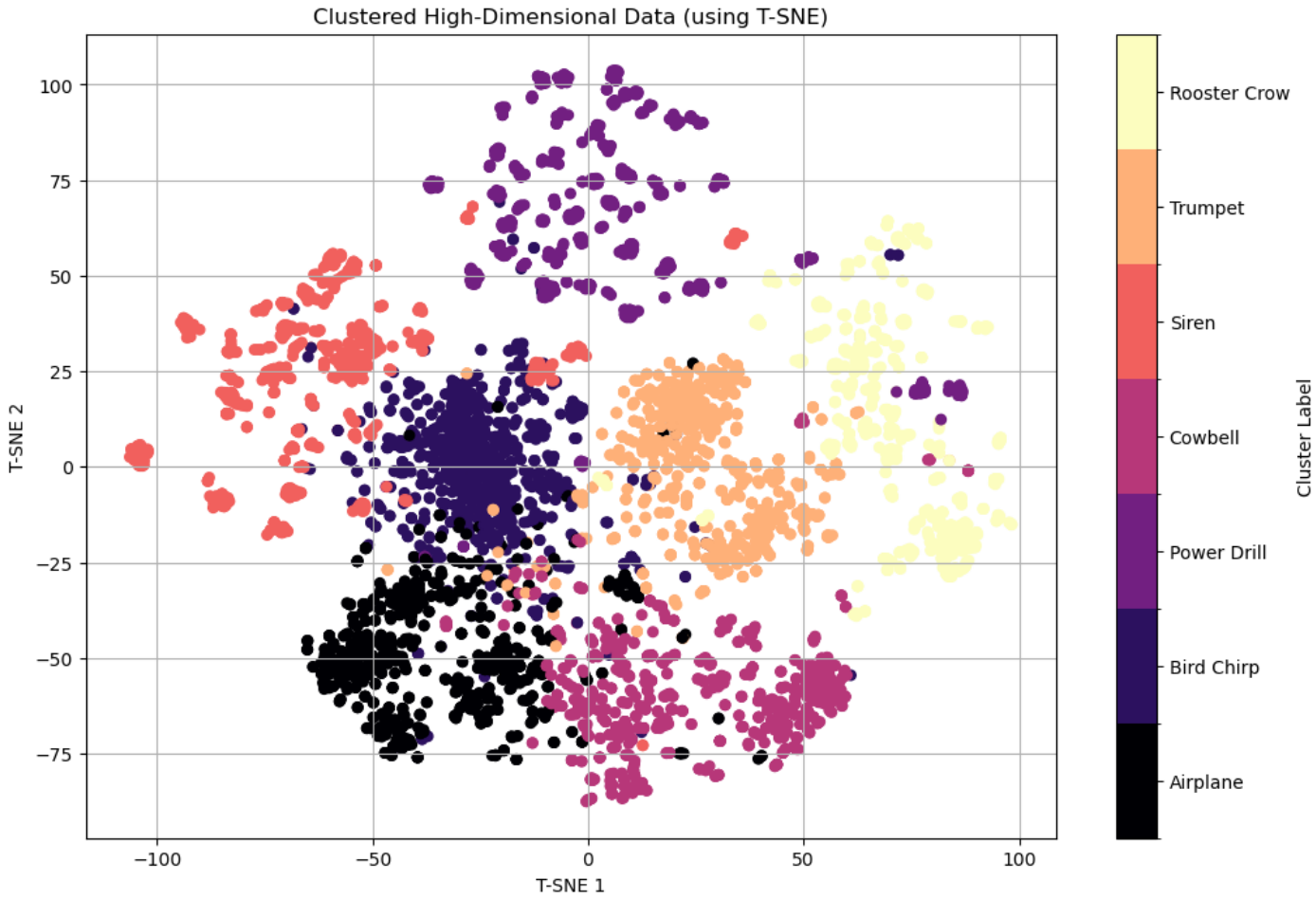
- Nonlinear relationships + combinatorial effects
- Trust the results?



Seemingly useful features: embeddings

How well do the chosen audio features group according to the discretized class labels? Do samples of the same class form tight clusters?

- Dimensionality reduction (T-SNE), then colored according to class-label.
- **Clusters clearly discernible**, some **overlap is to be expected**.



Audio Features

Which subset of audio features did you select for your final classifier? Describe the selection process and the criteria you used to make your choice

- Decided to play it safe and use all four features: **‘embeddings’, ‘melspectrogram’, ‘mfcc’ and ‘contrast’**.
- Unsure whether to trust the Random Forest, small-scale tests showed similar/slightly higher performance using all four.
- **No context frames, too computationally complex.**

Did you apply any preprocessing to the audio features? If so, explain which techniques you used and why they were necessary.

- We settled for **framewise mean/std normalization**, also tested global mean/std normalization using training set statistics.
- Ensures consistency in the scale of data -> **helps performance and convergence**. Predictions should not depend on e.g. volume.