Song Analysis and Recommendations

Vijay Ravuri - DSI Capstone

Outline

Background

- 8000 Songs, 30 second clips of each song
- Data gathered is all royalty free and publicly available (Free Music Archive)
  - That means no one has heard of these artists (except Charles Manson)
- Even genre distribution
  - Electronic, Instrumental, Folk, Rock, Hip-Hop, Pop, Experimental, International
- Genre is defined as the “top-genre”
  - That doesn’t mean too much
  - Songs can easily be two or more genres in reality

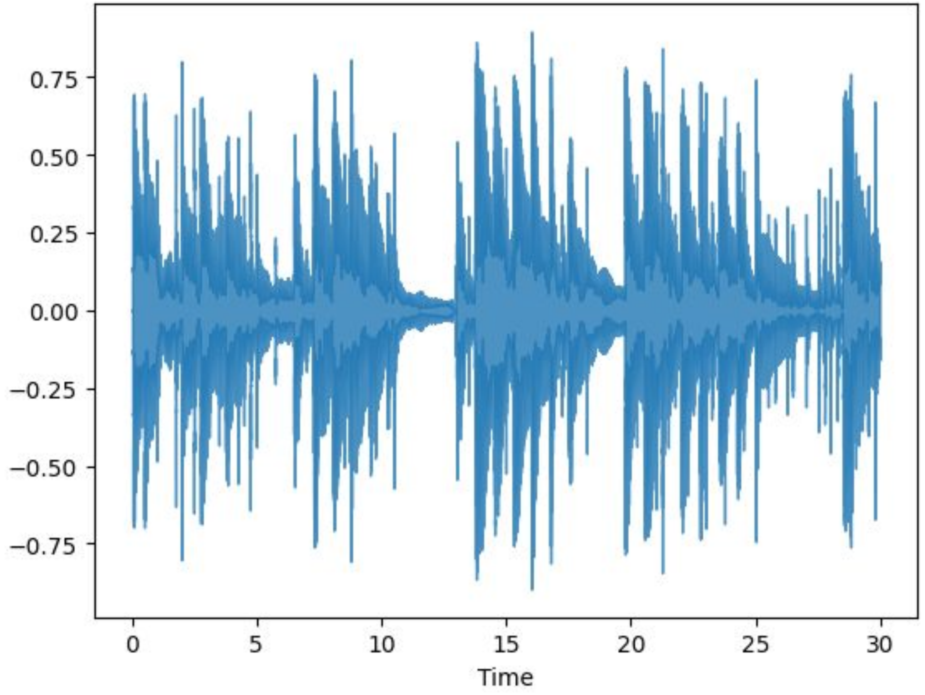
What do songs look like?

Common method: Waveplot

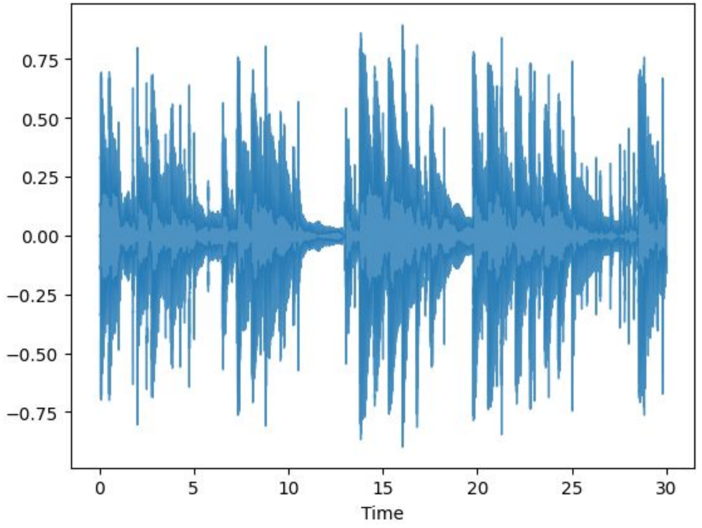
Krok by Blue Dot Sessions

Instrumental

Doesn't tell us that much about the song



Linear Algebra



+

Source: Wikipedia

Mel-Spectrogram

We can use Fourier Transforms to transform that waveplot into a mel-spectrogram

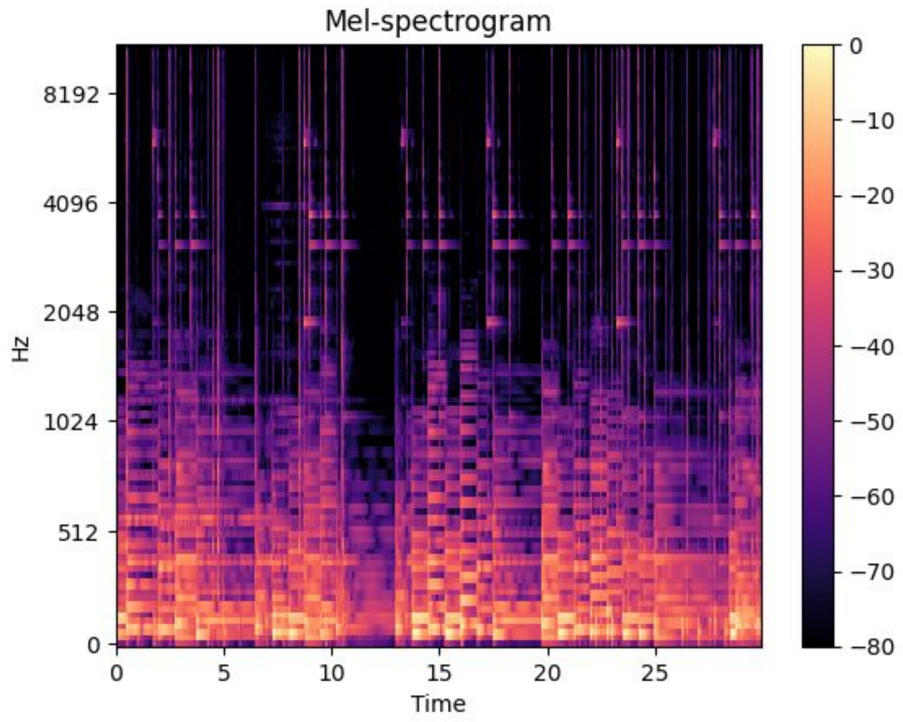
Maps songs into a format that’s easier to understand visually

Intuitively, this is a heatmap

30 second song → 1291 units long

128 frequency ranges (mel-bands)

Log-Scale for frequency (Hz)!



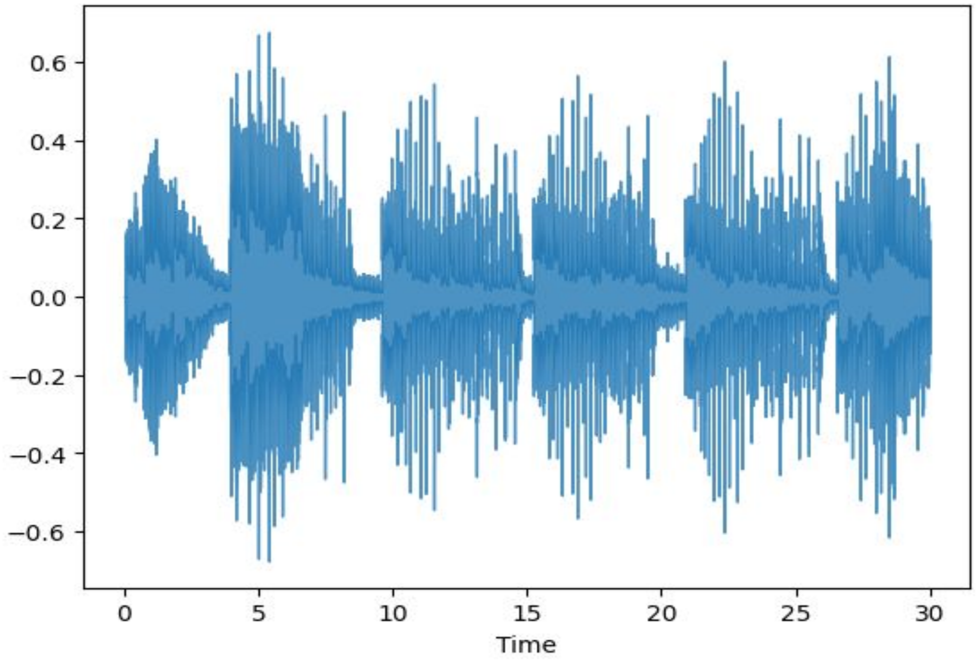
Comparing Songs

Memories by Psychadelik Pedestrian

Electronic

Very different shape to waveplot

(still not very useful)



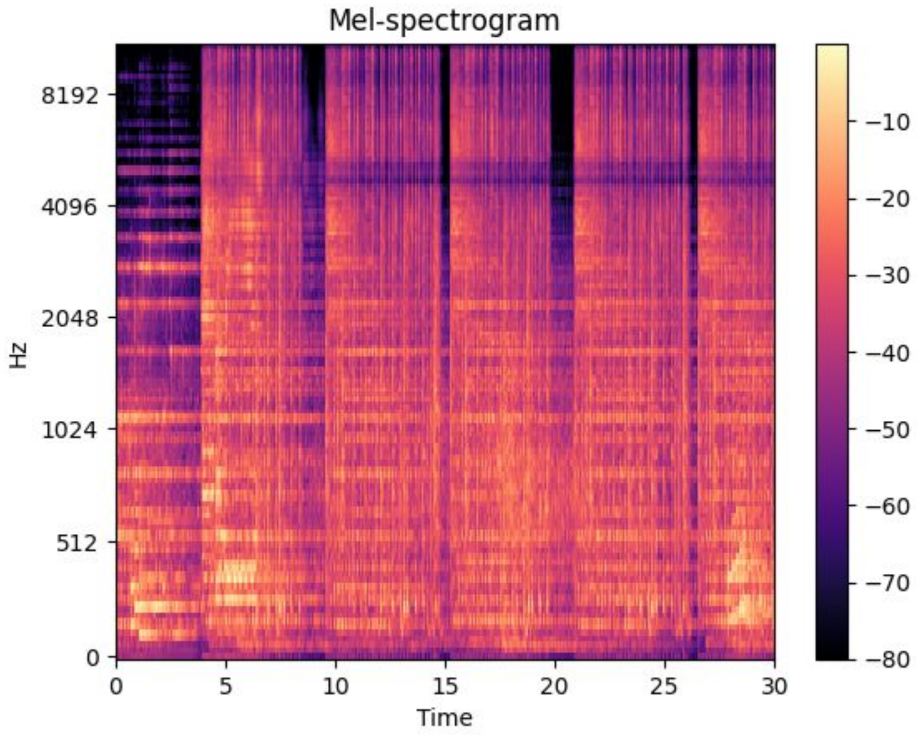
Electronic Spectrogram

Wildly different spectrogram

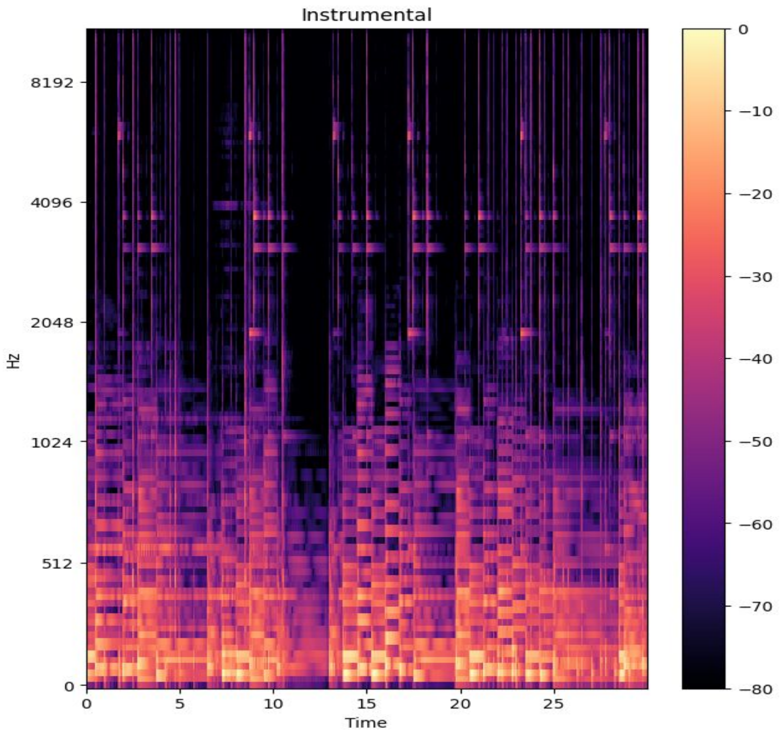
Everywhere, all the time

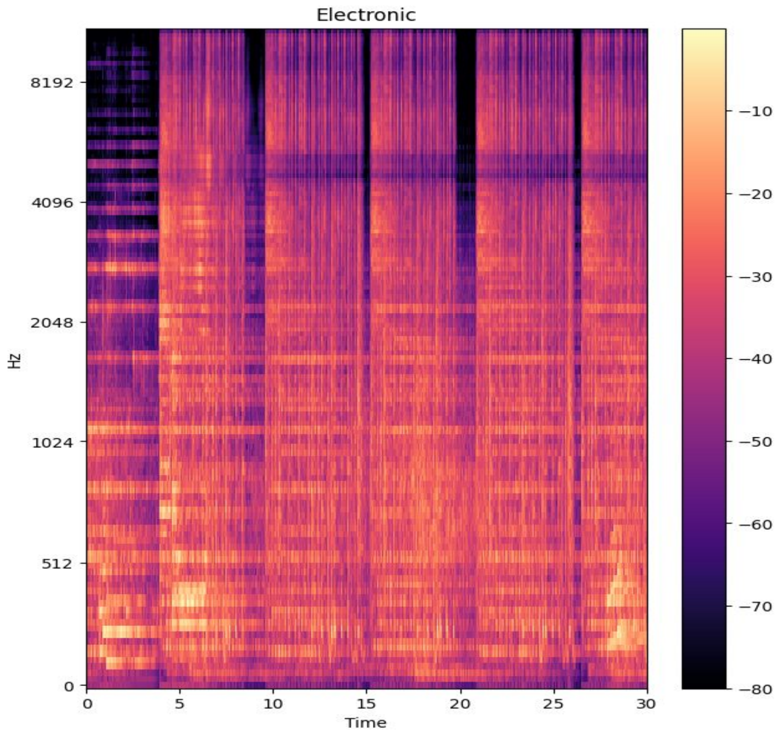
Much less range in loudness

Sharper changes in frequencies across time



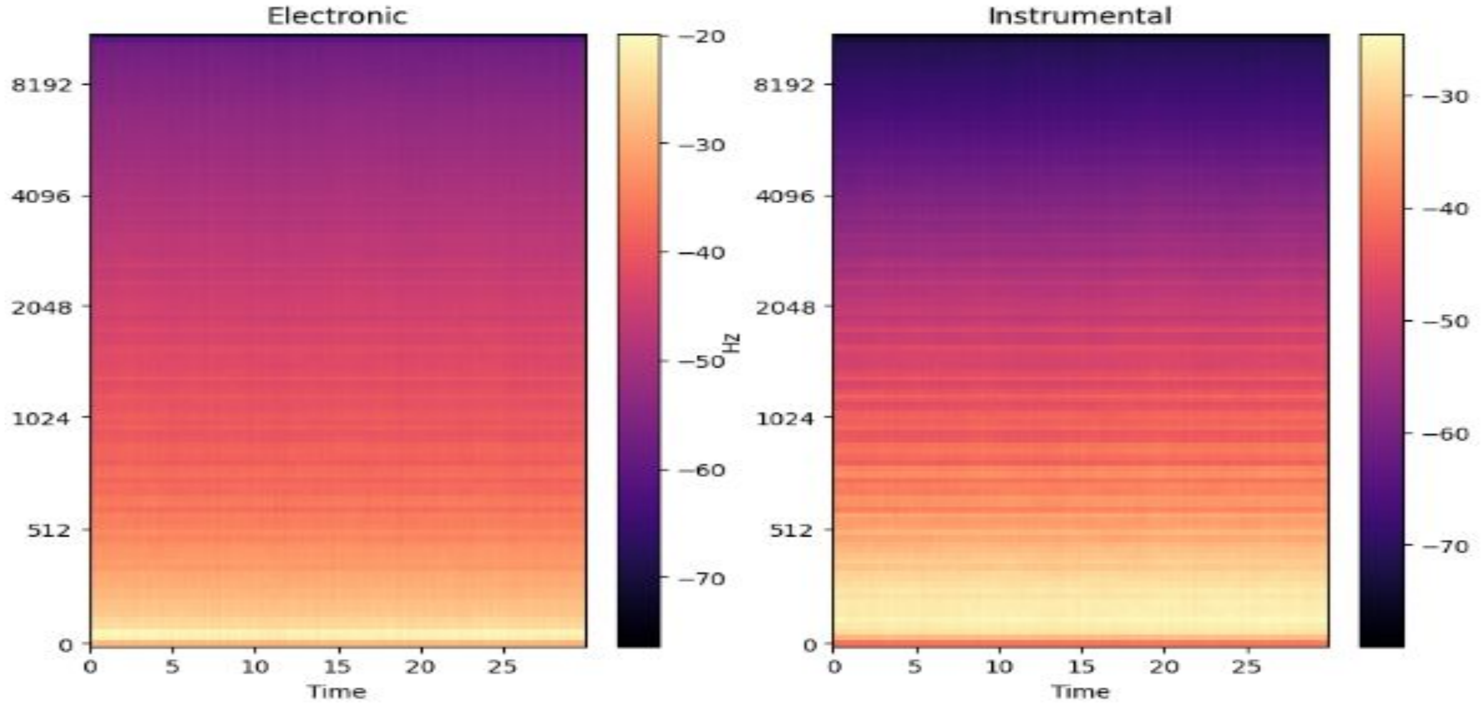
Compare the songs





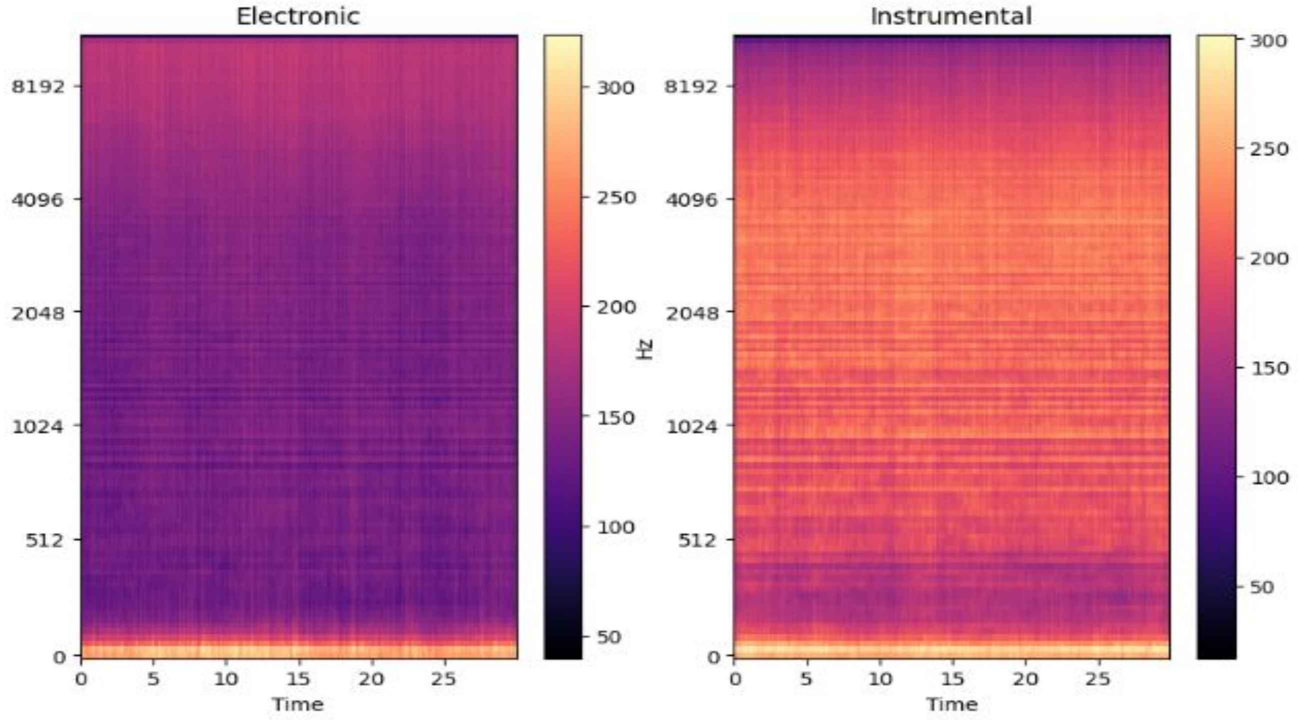
Genre comp (Mean)

Not too clear difference between the genres (low genre to genre variance)



Genre comp (Variance)

Very different here! Not too useful sadly



CNN

By and large, a failed attempt

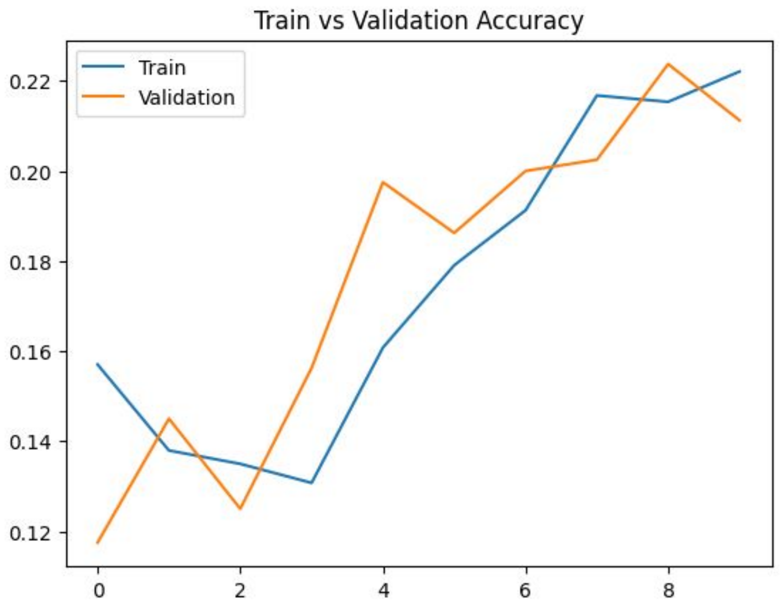
Took 90 minutes to train 10 epochs

Scored a spectacular 20.6% accuracy

2.43 Categorical Cross-Entropy (Log Loss)
(it’s literally log loss)

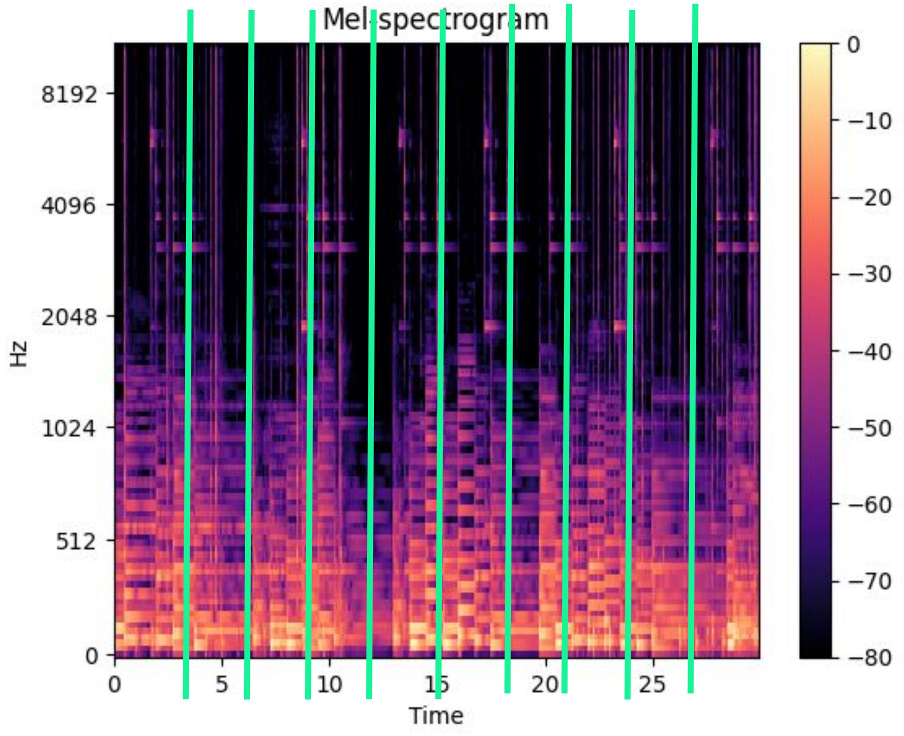
Baseline is 12.5% so 65% increase!

This approach has merit, I just wasn’t able to reach the point where it does



Reducing Complexity

1. Start with 1 song being 1291x128
2. Cut song into 10 even segments (10x129x128)
3. Take the mean value for each frequency range within these segments (10x1x128)
4. Take the standard deviation for each frequency range within these segments (10x1x256)
5. Apply PCA!

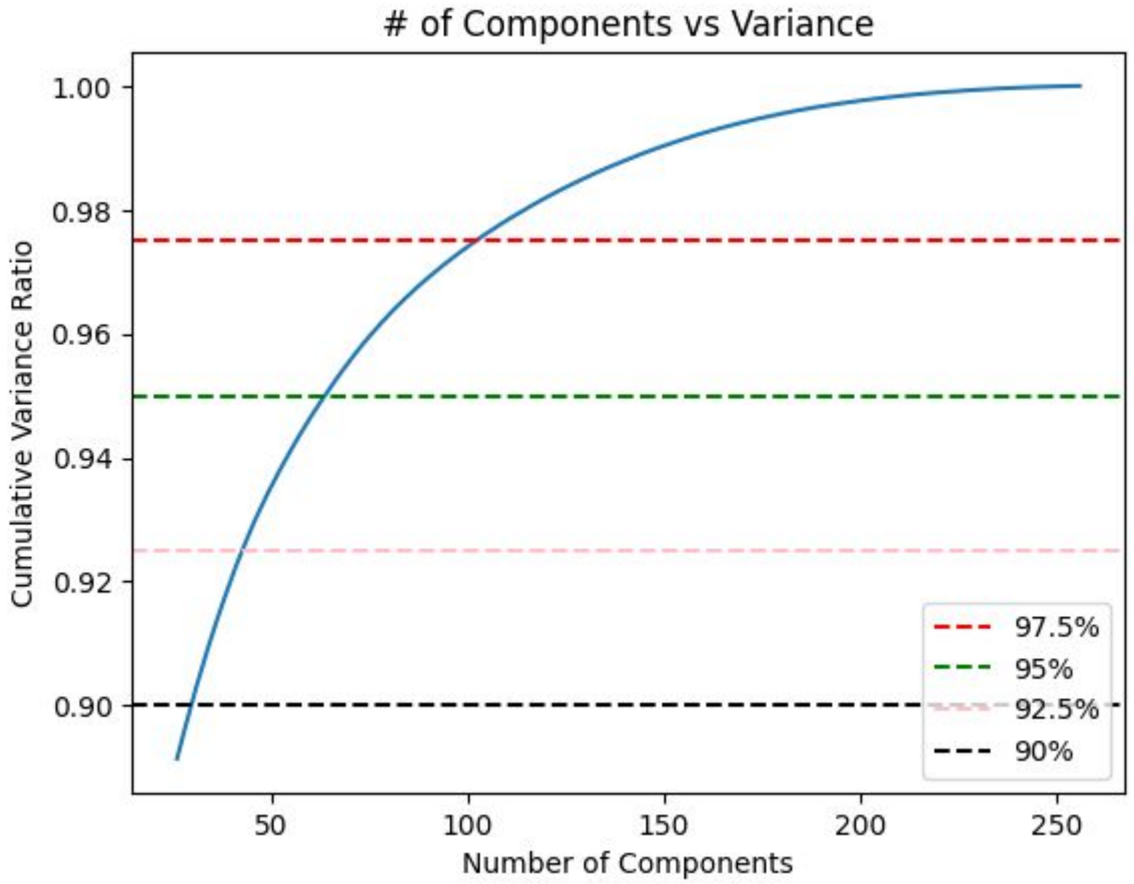


PCA

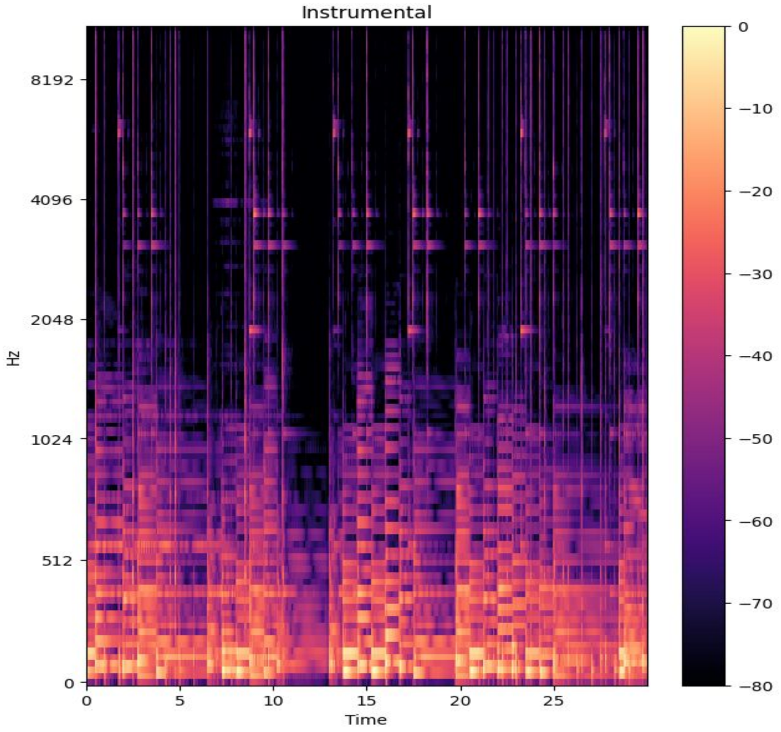
Main goal: Dimension Reduction

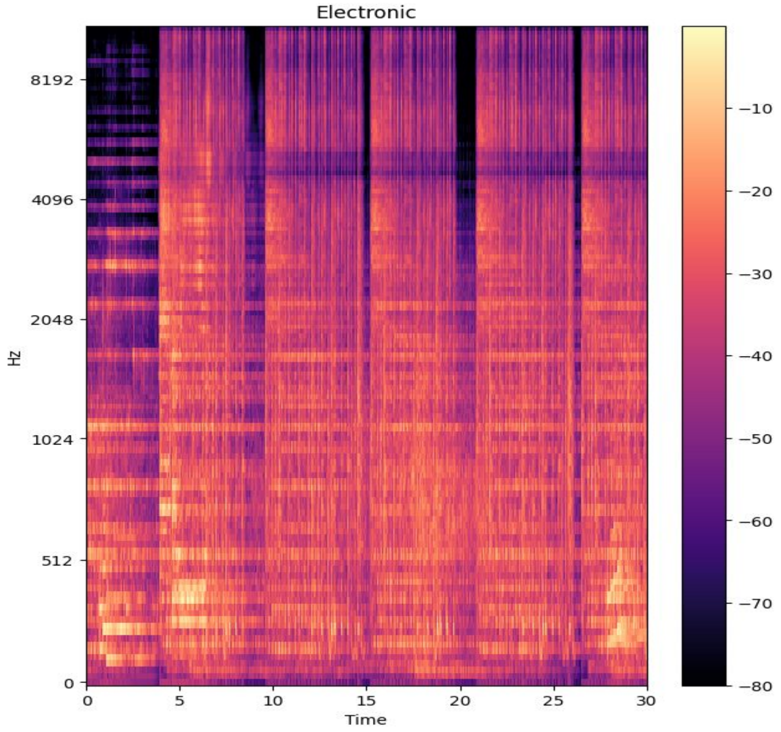
Secondary Goal: Spread our data

Used 100 principal components as the cutoff



How exactly does that work?





Now as decomposed components

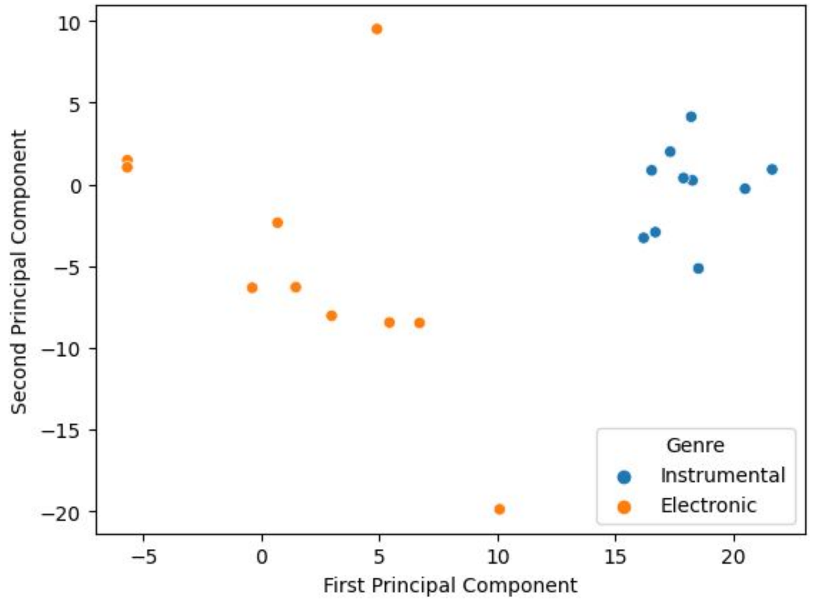
Each song is now represented as 10 3-second segments

Clear separation between the songs

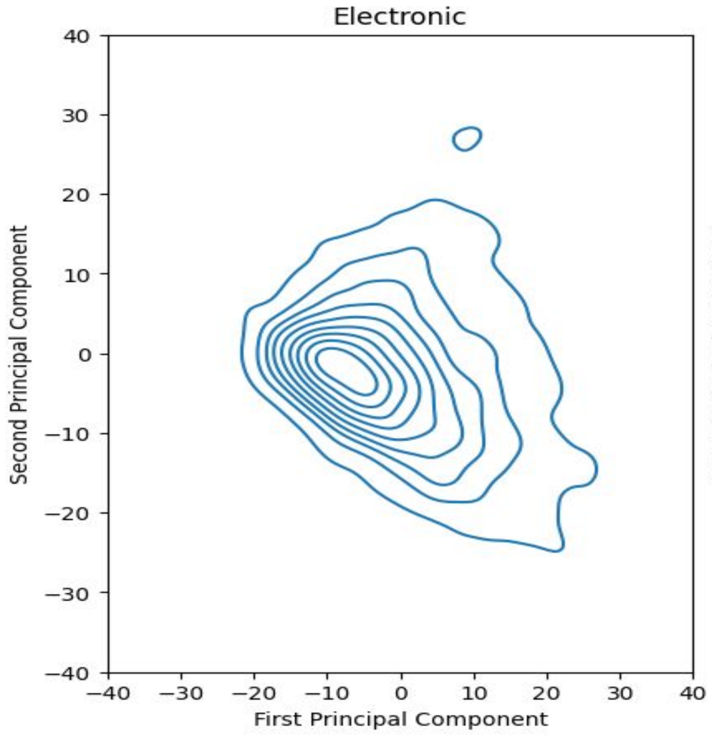
The songs themselves have their own unique spread

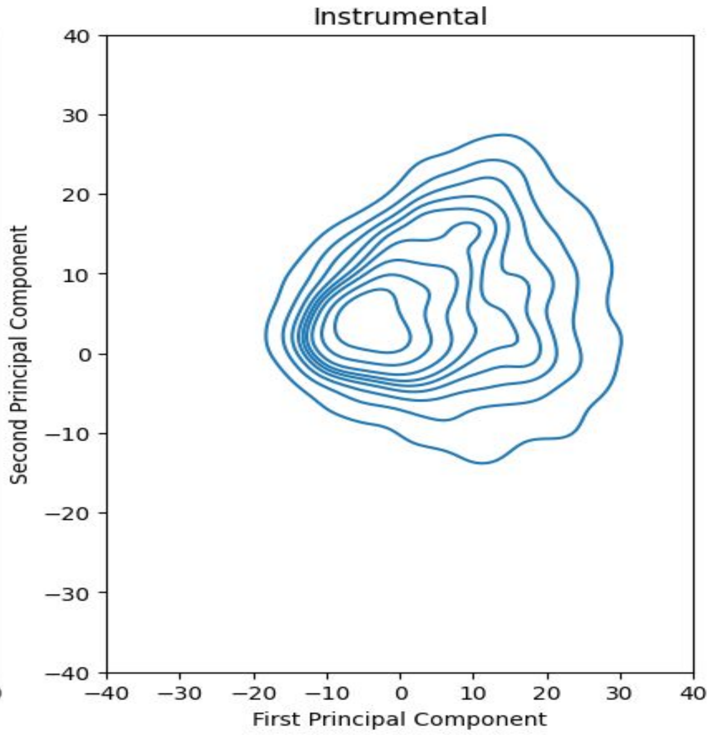
Instrumental song is more densely clumped than Electronic

See some evidence of high song to song variance



Genre Distribution





Modeling

Model of choice: Light GBM

Gradient-Boosted model

Lots of small decision trees, subsequent trees are fitted on the error of the previous tree

Again 12.5% baseline

Accuracy of….. 42.3%!

Recall and precision are identical

Performance Across Genre

Note the scale on the color bar!

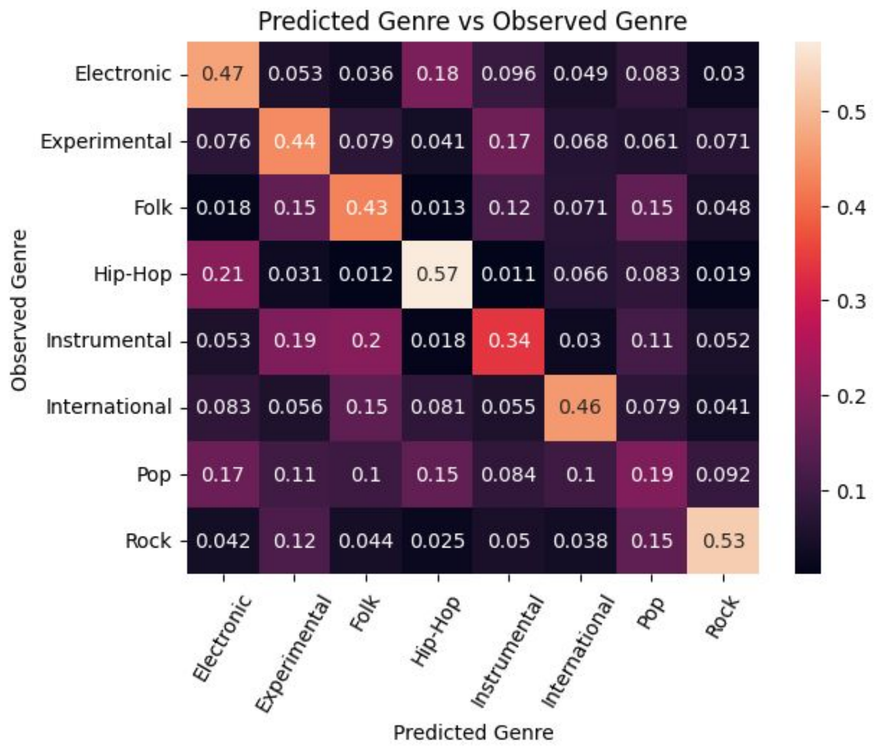
Model struggled a lot with Pop

Did very well with Hip-Hop

Instrumental was weaker as well

Overall, not the worst performance (beats CNN)

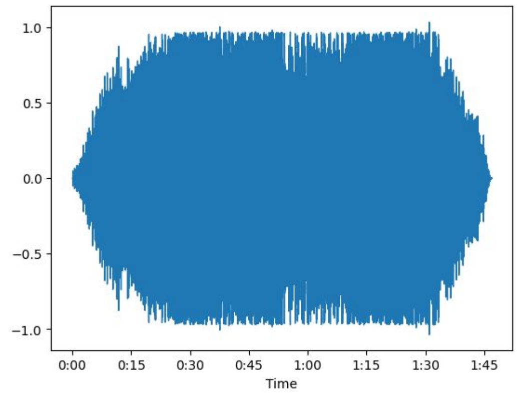
Genre overlap is a big problem (Hip-Hop, Electronic)

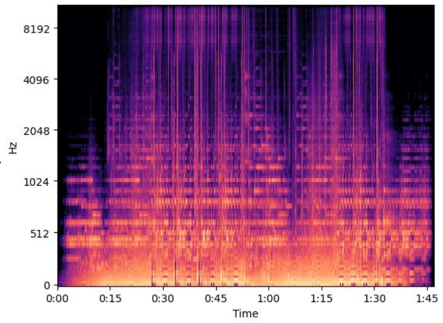


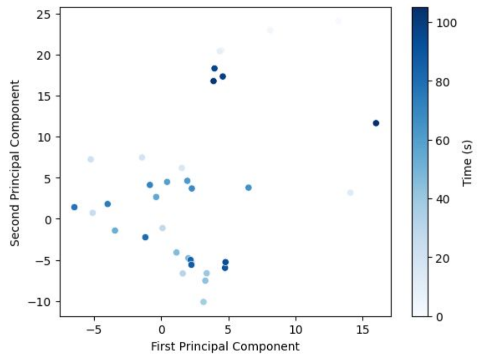
Observed \ Predicted	Electronic	Experimental	Folk	Hip-Hop	Instrumental	International	Pop	Rock
Electronic	0.47	0.053	0.036	0.18	0.096	0.049	0.083	0.03
Experimental	0.076	0.44	0.079	0.041	0.17	0.068	0.061	0.071
Folk	0.018	0.15	0.43	0.013	0.12	0.071	0.15	0.048
Hip-Hop	0.21	0.031	0.012	0.57	0.011	0.066	0.083	0.019
Instrumental	0.053	0.19	0.2	0.018	0.34	0.03	0.11	0.052
International	0.083	0.056	0.15	0.081	0.055	0.46	0.079	0.041
Pop	0.17	0.11	0.1	0.15	0.084	0.1	0.19	0.092
Rock	0.042	0.12	0.044	0.025	0.05	0.038	0.15	0.53

Recommender System

1. Get user input song (Any length!)
2. Break song down into 3-second segments
3. Perform the same aggregation and PCA as before
4. Compute Cosine Similarity for each segment against our dataset (slightly de-weight first ~9s)
5. Take the top 10 most similar segments for each segment of the user's song
6. Return the top 5 songs by sum of their similarities







Streamlit Demo

Conclusion & Future Work

- Audio data is hard to work with
  - Computationally and conceptually
- Applying neural networks to actual problems is incredibly difficult
  - For future work on this type of data, transfer learning or cloud computing are the way to go
  - On a personal scale, neural networks are a waste of time
- Genres have a lot of overlap!
  - Low genre to genre variance
  - High within genre variance
- Tons more to be done in terms of building a more complex recommender system
  - Weighting the recommendations differently to balance frequency of similarity and amount of similarity
  - Different weights for time segments?
    - De-weighting first 9s was a rudimentary attempt
- Limiting genre options might help with making a better model or more accurate recommendations

Q&A