Music Genre Classification and Hit Prediction

CSE 575
Group 12

Keval Shah 1225553979
Rithik Goyal 1225572699
Shashank Karanth 1225314194
Vidushi Raturi 1225495622
Bitaan Chakrabarti 1225499119
Aman Peshin 1225476655

Problem Statement

- Given an audio file, extract audio features and classify the genre that it belongs to.

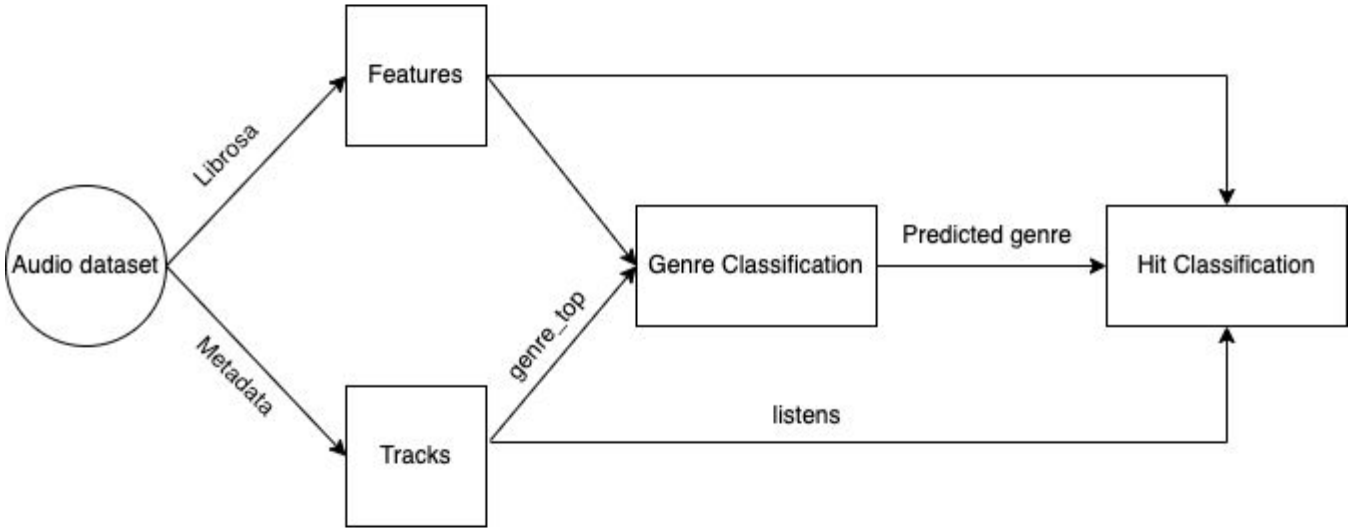
- Consequently, predict whether it is/was/will be a hit based on other songs belonging to that genre.

- (Future Work) - given the features of the sample song, we want to predict the actual features that would make it a Hit song according to the analysis of our models.

Motivation

- Music creation and Hit generation has been a need since forever.
- In the current situation, there are a lot of intangibles which are not in the creators' hands, that make music a Hit or not.
- We want to give the power to create the best music possible, back to creators, based on an analysis about the musical features of their music.
- Music is highly subjective and the abstraction of the Music aggregators and Record Labels doesn't really help the creators in any way.
- People perceive music differently and subtle nuances in tone, pitch, etc. differentiates it.
- Thus it is important to categorize music into different genres.

Methodology



Comparison with Existing Work

The best results were seen when using Spotify Data (features) - for Hit prediction and Genre prediction.

- Song Hit Prediction: Predicting Billboard Hits Using Spotify Data - https://doi.org/10.48550/arXiv.1908.08609 (88% - accuracy)
- HITPREDICT: PREDICTING HIT SONGS USING SPOTIFY DATA - https://cs229.stanford.edu/proj2018/report/16.pdf (82% - accuracy)
- Genre Classification of Spotify Songs - https://cs229.stanford.edu/proj2017/final-reports/5242682.pdf (82% accuracy)

Similar work was done with very similar results as that of ours:

- Music Genre Classification using Spectral features - https://github.com/Pedrohgv/Music_Genre_Classification (Accuracy measure - 65%)
- MULTI-LABEL MUSIC GENRE CLASSIFICATION FROM AUDIO - https://arxiv.org/pdf/1707.04916v1.pdf (88.8% from the AUC plot)

FMA Dataset

Data Set Information:

- Audio track (encoded as mp3) of each of the 106,574 tracks arranged in a hierarchical taxonomy of 161 genres. It is on average 10 millions samples per track.

Attribute Information:

- Nine audio features computed across time and summarized with seven statistics (mean, standard deviation, skew, kurtosis, median, minimum, maximum):
- Features: Chroma, Tonnetz, Mel Frequency Cepstral Coefficient (MFCC), Spectral centroid, Spectral bandwidth, Spectral contrast, Spectral rolloff, Root Mean Square energy, and Zero-crossing rate.

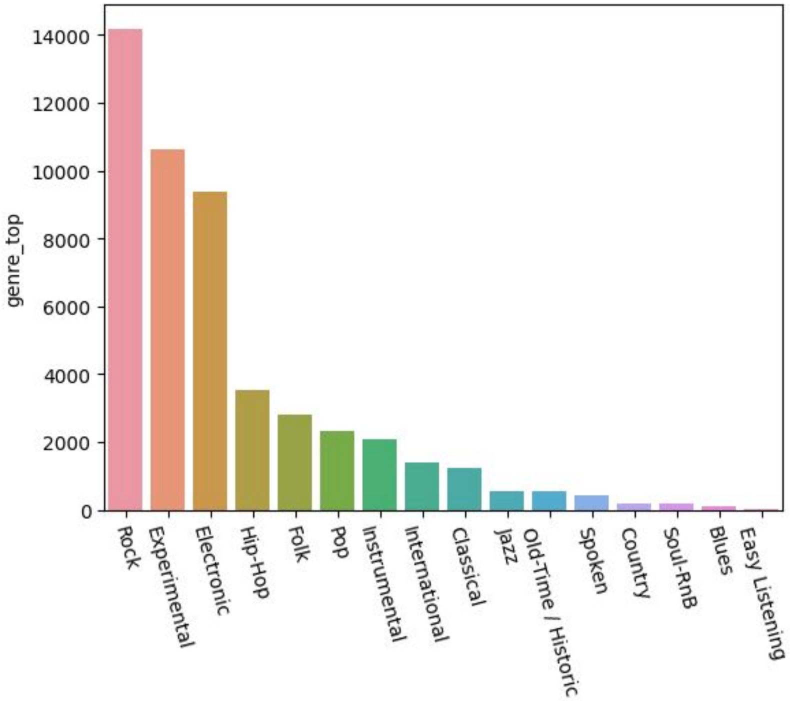
Music Features

- MFCC
  - It is based on a logarithmic scale and is able to estimate human auditory response in a better way than the other cepstral feature extraction techniques.
- Chroma
  - It is a powerful tool for analyzing music features whose pitches can be meaningfully categorized. They capture harmonic and melodic characteristics of music while being robust to changes in timbre and instrumentation.
- Spectral Rolloff
  - Spectral Rolloff is the frequency below which a specified percentage of the total spectral energy
- Zero Crossing Rate
  - Zero-crossing rate is a measure of the number of times in a given time interval/frame that the amplitude of the speech signals passes through a value of zero. It is a key feature to classify percussive sounds.

Genre Classification

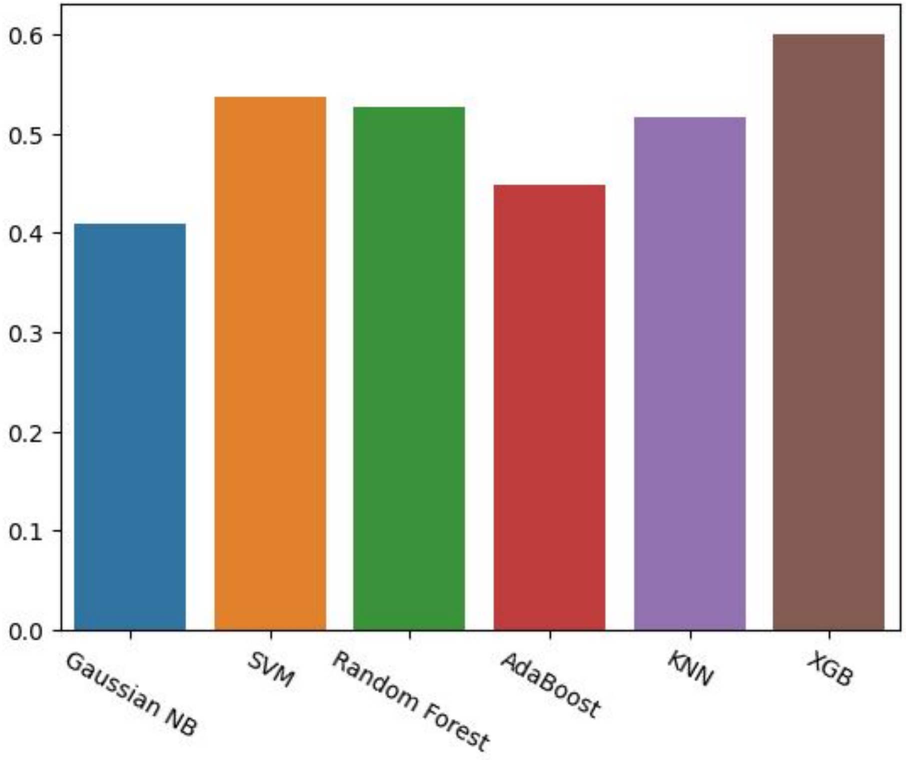
- Genres are categories used to distinguish between various kinds of music.
- Features serve as the input to pattern recognition systems and are the basis upon which classifications are made.
- From here, you can perform other tasks on musical data like beat tracking, music generation, recommender systems, track separation and instrument recognition, etc.
- Music analysis is an interesting challenge in the field of Data Science.

Data



- Highly imbalanced data.
- We take a balanced subset of 2000 songs per genre from the top 7 genres.
- We only consider the MFCC feature for genre classification, which has 140 columns in total.

Results

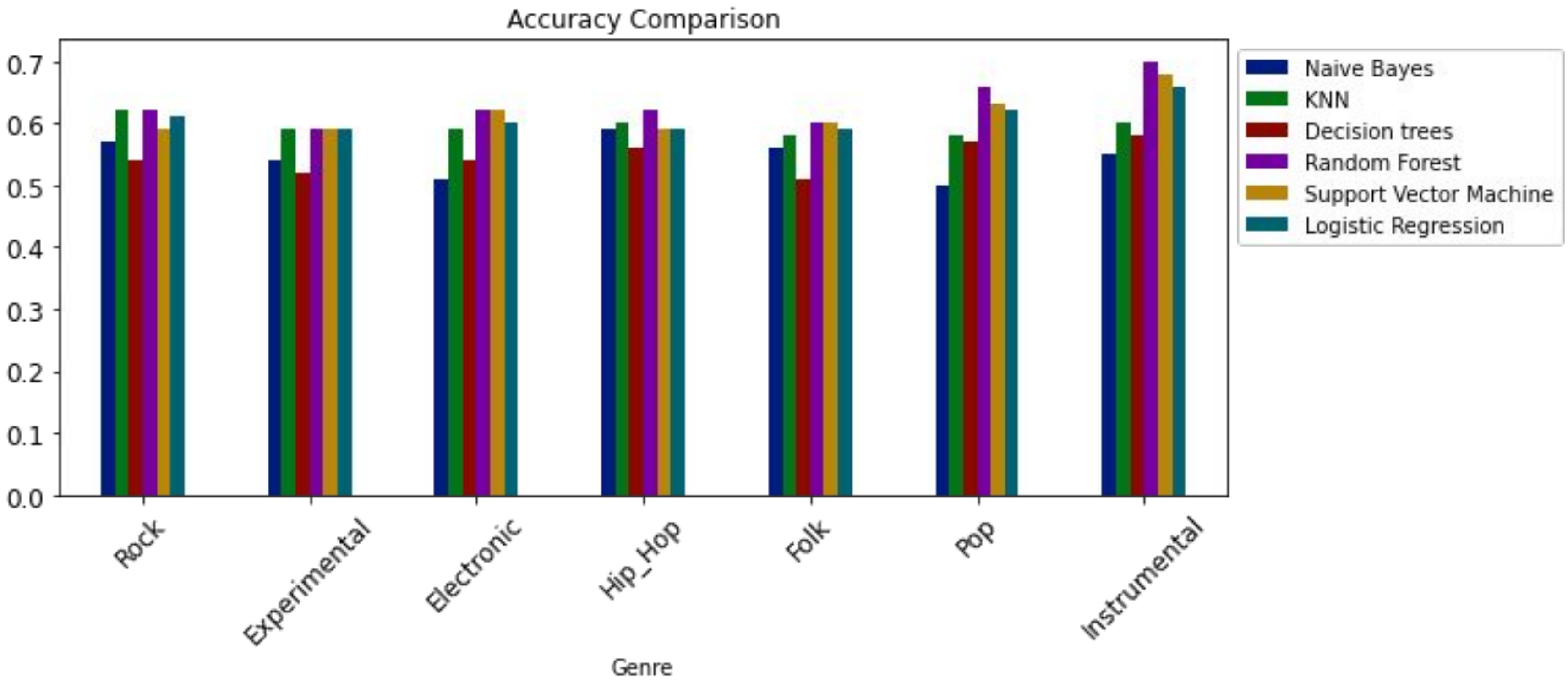


- Best model gives close to 60% accuracy.
- Hyperparameter tuning does not improve accuracy significantly.
- Accuracy is low.

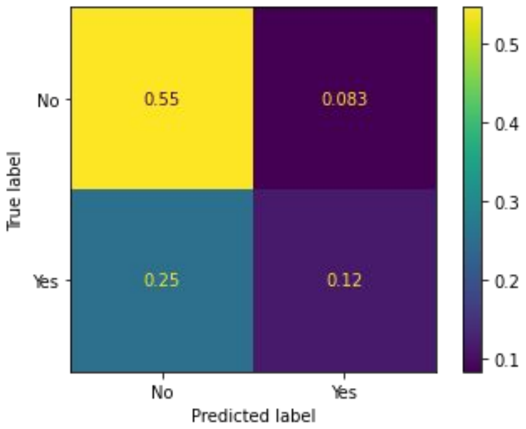
Hit Prediction

- We used the ‘Listens’ column/ label to emulate the “Popularity” of music so as to threshold the music as Hits or not.
- We made multiple different models - one for each genre, because different genres have different parameters that make it a ‘Hit’
- We used more feature sets compared to just ‘MFCC’ used for Genre prediction, since the Hit classification prediction based simply on the ‘MFCC’ feature performed significantly poorly than using a model with a combination of features - ‘MFCC’, ‘Chroma’, ‘Spectral Rolloff’ and ‘ZCR’ - which have been discussed earlier.

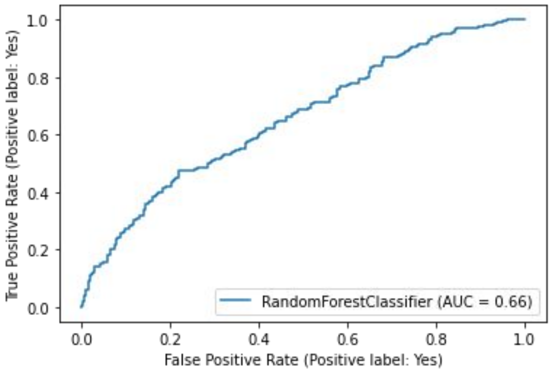
Accuracy

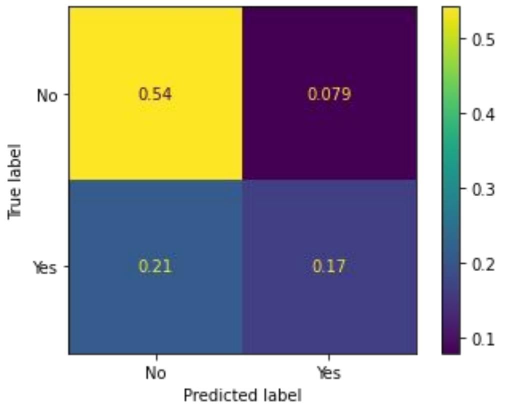

Accuracy Comparison
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Rock
Experimental
Electronic
Hip_Hop
Folk
Pop
Instrumental
Genre
Naive Bayes
KNN
Decision trees
Random Forest
Support Vector Machine
Logistic Regression

Results

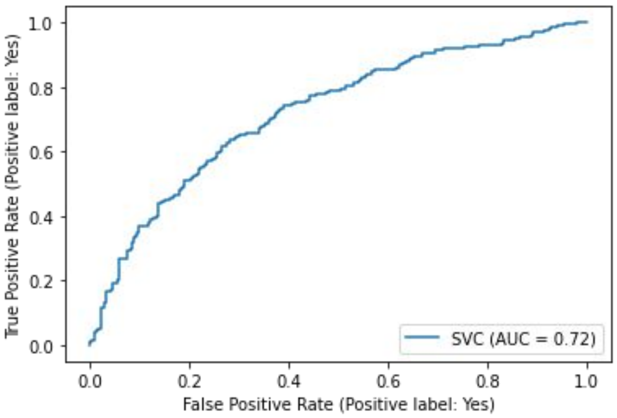


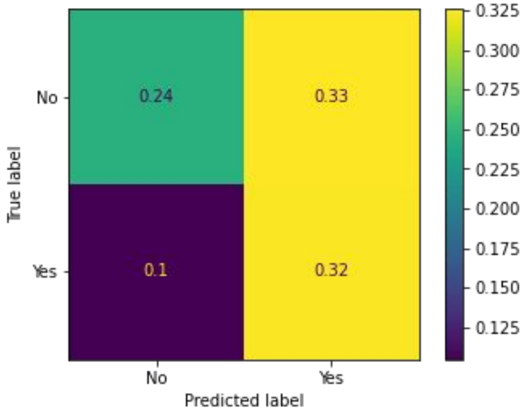
Pop



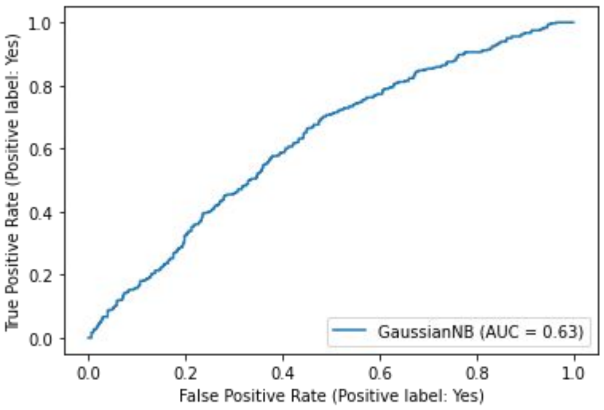


Instrumental





Folk

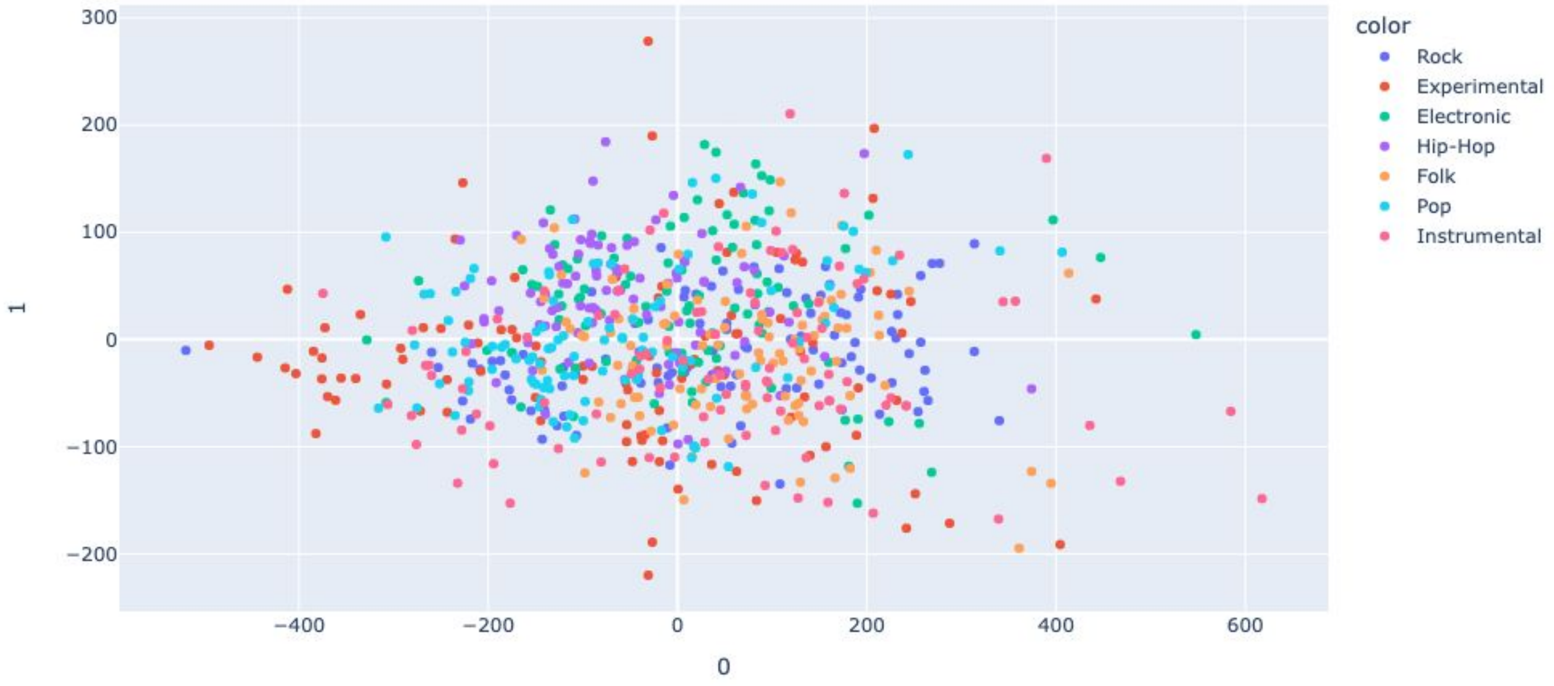


Challenges / Lessons Learned

- Low accuracy for both tasks. Hyperparameter tuning also did not change accuracy significantly. Why?
    - ❑ Overlapping clusters. The variance of values of features for different classes is very low.
    - ❑ The given features/data are not good measures to predict genres/hits.
    - ❑ Statistical models are too simple to understand such complex data?

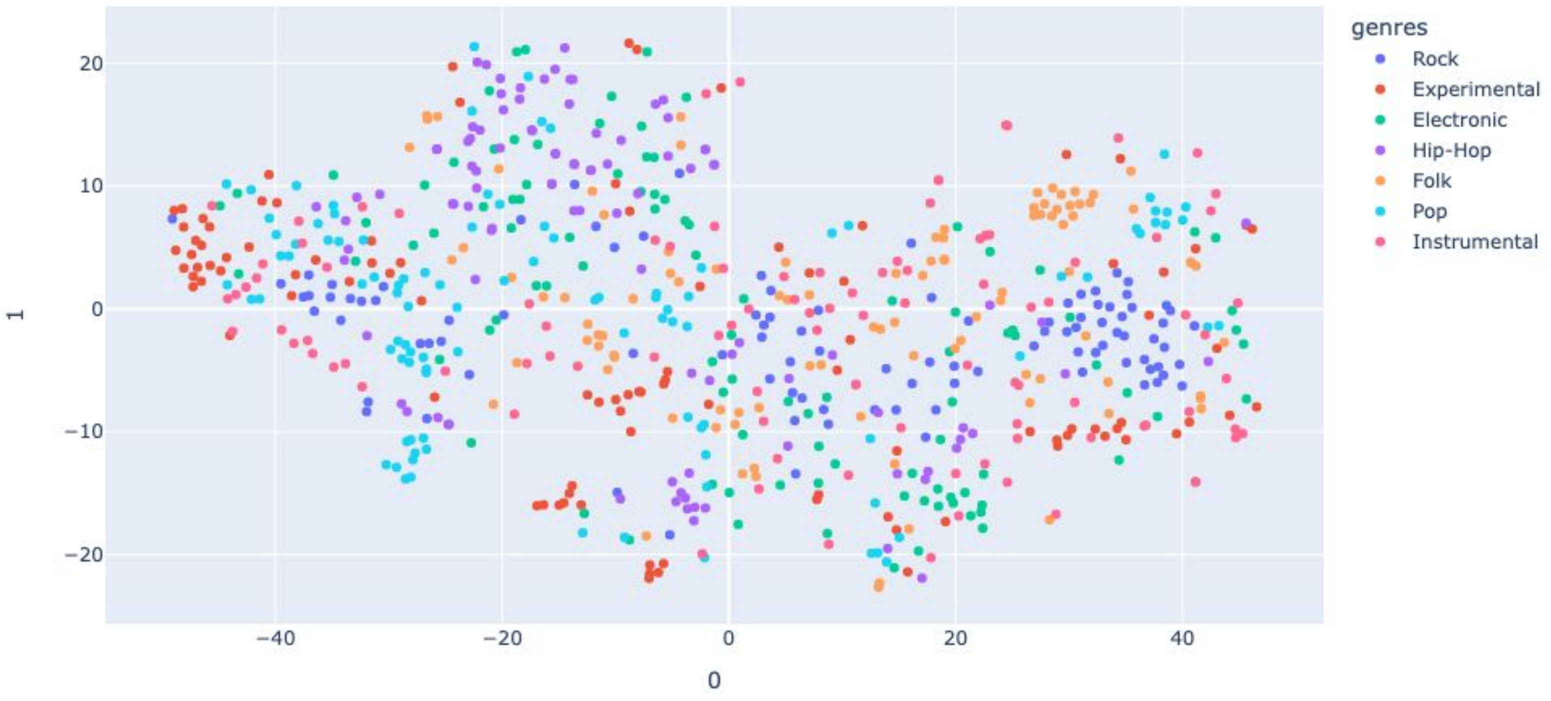
Visualization of Data

PCA for dimensionality reduction



Visualization of Data

t-SNE for dimensionality reduction



Next Steps / Conclusion

- Additional tuning for our base models to reach 70% accuracy.
- Mapping/ Combining spectral features to mimic echonest features. Which in turn could give us better classification accuracy with out of the box models.
- Using Deep Neural Nets to learn this mapping between the two feature sets.
- Stretch Goal: Reconstructing audio using Griffin-Lim algorithm.