## Notes on Progress

## December 11th

6:41 - Genre filtering is done

2:09 - got the recommender to work in the Final Product, need to run a quick errand but I'll try to add genre filtering

1:05 - I got the Feature Extractor to work in the Final Product Colab

12:33 - It looks like I need to convert the Functioned\_echonest\_extractor into a .py file to import it

11:54 - I'm going to try to convert all the code so that it works with functions that can be called on by the final google colab

11:30 - So, today I'm going to try to clean everything up so all the working code goes in a single Google Colab

* Note: Because we weren't able to implement the other models due to time restraints so instead of using the 60 cluster recommended by the elbow method I'll use 250 cluster because that gives like 25 recommended songs
* The problem with DBSCAN was that even after cleaning the dataset more and messing with the parameter, most of the data points were still labled as outliers

## December 10th

5:18 - Apprently the optimal number of clusters is 60

From now on we are using echonest\_mod4

3:24 - It works well enough for now, maybe I can add a tab that lets the user select what genre the song is in.

* An I might have to try to find the optimal number of clusters

2:50 - I think scalling the tempo might have broken the code so, I'll test it and if it doesn't work out I'll train another K-cluster and and use the original feature extractor, but make sure the application for k-cluster removes the Liviness

* The tempo change is definetly what is breaking the code. Welp. I'm going to clean up the shared drive then I need to make another k-cluster model that has the default tempo, but does not have liveliness. Also, I'm going to keep the orginal feature extractor and just make sure that when the new data is implemented that it checks to see if it has liveliness and remove it if it does.

2:30 - Going to try to apply the new model to the Johnny B. Good data

1:54 - creating a new k-cluster trainer using the echonest\_mod5 and johnny be good

Going to extract the features from "Johnny Be Good" inside of Echonest Feature Extractor version 2, but removing liviness and scaling the tempo

1:31 - For echonest\_mod5, I've removed the Liveness, and I changed the tempo to be from 0 to 1 where 15 bpm is 0 and 1 is 300

1:16 - <https://developer.spotify.com/documentation/web-api/reference/get-audio-features>

The link shows what the audio features are

* Going to remove liviness

12:57 - Going to try to remove tempo as the rest seem to be from 0 to 1

12:44

* Working on trying to get DBSCAN to have less of the datapoints labled as noise

1. eps (epsilon):
   1. eps is a positive floating-point value that represents the maximum distance between two samples for one to be considered as in the neighborhood of the other.
   2. It defines the radius within which the algorithm looks for neighboring points to form a dense region.
   3. Increasing the eps value allows for larger neighborhoods, potentially merging clusters and considering more points as part of the same cluster.
2. min\_samples:
   1. min\_samples is an integer that represents the minimum number of samples (data points) required to form a dense region (core point).
   2. If the number of samples within the eps radius of a point is less than min\_samples, that point is considered as a border point.
   3. If the number of samples within the eps radius is greater than or equal to min\_samples, the point is considered a core point, and a cluster is formed.

## December 4th

5pm - tried with 500 clusters. It seems alright but I'm going to bring it down to 250 to give us some more wiggle room as it only gave us 10 songs

* It basically gave me a completely different set of songs

4:42 - the elbow method doesn't seem to be working. I'll try again k-cluster with 100 clusters now with the new echonest

12:08: Pretty sure I removed all the ones I needed to from tracks\_mod, now I just need to do the same to echonest

* I filtered it, but there weren't many that were removed

11:41 - Going to remove the experimental rows from echonest

* Also removed the ones with empty strings or numbers

## Decemeber 3rd

* K-nearest

Further Clean up dataset to remove experimental genre to improve dataset

4:55 - ~~Try random forest~~  Trying DBSCAN

* The DBSCAN is only creating 11 clusters, it might be worth it to try to have this many clusters in the k-cluster model
* The model is considering most of the songs outliers which means we'll probably have to tweak some of the parameters, but I'll test it anyways
* Yeah, "Stressed Out was " labled as an outlier

## Decemeber 2nd

4:13 - Trying Autoencoder

Making Heirarchical Cluster

* Didn't work because the new\_data is only one row, and hierarchical clustering is typically used for datasets with multiple samples

## November 30th

8:02

Cleaned up the Echonest Feature extractor code so that it should be easier to work with in the future

## November 29th

9:08

It worked

6:48

Working on k-cluster

* echonest\_mod3 one hase
  + Acousticness
  + Danceability
  + Energy
  + Instrumentalness:
  + Liveness:
  + Speechiness
  + Tempo
  + Valence

6:28pm

Finally going to make echonest\_mod2

* Going to remove values in hotness and currency that are zero (success)

4:41pm

~~Going to try using that features code from the FMA~~ Actually, let me try again. I think the problem is with the URI

9:35:

Luis: Going to look at spotipy. It is still taking way too long, as in I've had it run for more than an hour and it didn't work

## November 28th

Jake: Working with extracting features from Features

* Feature extraction with a full length songs uses too much ram and crashes the program

Luis: Going to mess with echonest

* echonest is now spodify apprently
* 5:14: I'll try out Spotipy!

## Novemeber 27th

3:29: I'm going to try to use features mod just to be safe, but it might not work with the below thing

2:35: I'm going to try to use the code from features.py to extract the features rather than what librosa does by default

2:30: As a reminder. "features\_mod2" is the one with the averaged out values

2:19

Same result as last time

2:12

It doesn't work at all. All I'm getting is white noise



For future refrence, the the first attempt with this only showed the song actually called "White Noise"

I'm going to try again but with 25 Clusters

10:47

The Model looks much better

10:12

Just ran the model with 50 clusters. Let's see how it goes

## Novemeber 24th

6:46

Made the new model with 100 clusters and applied it. But it failed terribly. either that or the track\_ids are not consisiten across the different files

I'l try again tomorrow but with fewer clusters. Let's say 50

01:32

I extracted the features from the song and turned it into a dataframe and it turns out that its shape is (8707, 5).

So what I'm going to try is find the average of these for each row of the features dataset: mfcc chroma mel contrast tonnetz

in the features dataset, so then for the actual song I will the average too

12:45

The kmeans cluster seems to have worked out well with 100 clusters. I'll try to implement it now

11:28

Apparently features has a cluster column, I'm going to see what happens when I remove it

10: 55

Stopping with the autoencoder, for now and going to try k-clustering with features

9:24

Continueing from tomorrow

## Novemeber 23rd

1:46

Going to try to save the model and test it with a song recommendation

12:54

Finally running the encoder model

12: 35

Tried graphing features cluster, but it took more than 25 mins so I interupted it.

11:17

For the features CSV, I'm going to use clustering the following clustering methods:

ChatGPT

"

**Autoencoders**:

Reason: Autoencoders are effective at learning compact representations of input data. In the context of music recommendation, they can capture intricate patterns and features in the audio data, providing a meaningful latent representation. Including multiple autoencoders with diverse architectures or hyperparameters allows the ensemble to capture different aspects of the data's complexity.

**K-Means Clustering:**

Reason: K-means clustering is a simple and interpretable algorithm that can group similar audio features together. By varying the number of clusters, you can explore different granularities of grouping. Ensemble over different cluster counts to provide a robust representation of the data's inherent structure.

**Hierarchical Clustering:**

Reason: Hierarchical clustering captures both global and local structures in the data, creating a hierarchy of clusters. It can be particularly useful if there are nested relationships in the music data. Ensemble over different linkage methods or distance metrics to ensure a comprehensive exploration of clustering possibilities.

"

10:51

genre\_mod changed to only have id and name of genre

10:46

Going to mess with Features CSV

I'll keep features the same for now as they all seem to be actually numbers from librosa

10:30

I've decided to keep temporal features as they generally refer to features that capture information about how musical properties change over time within a song.

## November 22nd

8:35: Reading the Paper

"We propose an 80/10/10% split into training, validation

and test sets to make research on the FMA reproducible."

"Permissive licensing. MIR research has historically

suffered from the lack of publicly available benchmark

datasets, which stem from the commercial interest in music by record labels, and therefore imposed rigid copyright.

The FMA’s solution is to aim for tracks which license permits redistribution. All data and code produced by ourselves are licensed under the CC BY 4.0 and MIT licenses.

"

8:21

I'm going to modify the features file we have

Also I need to double check the paper, but I'm pretty sure that the track\_id is correlates to the same song across the files

Features looks like a pain, so maybe not until I read more.

Let's look at echonest instead

For echonest, I'm removing the metadata (mostly artist information)

Also removed:

* For social features, I deleted everything except for as they should hopefull remove any overly obscure songs

From Book

## November 21st

12:30: Going to look at Librosa next because Echonest has some columns with whose names are numbers

Quote from paper about our database "Diversity is another issue. As suggested by Figure 6, this collection is biased toward experimental, electronic, and rock music. Moreover, it does not contain mainstream music and few commercially successful artists"

12:12pm

For tracks\_mod I removed:

* Composer
* date\_created
* date\_recorded
* favorites
* Removed tags as it seem the majority of songs lacked them
* publisher
* lyricist
* Number
* license
* listens
* language
* information
* interest

Maybe let's not use the tracks csv it seems to me that the only useful information is the genre

11:54 AM

Trying to manually fix dataframe

* Removed Artist and Album information
* Removed bitrate
* Creatinging a file with all track information and a modified track information that is more curated

Important Note: Songs only go up to 2017

From Echnoest removed

* album\_date
* album\_name