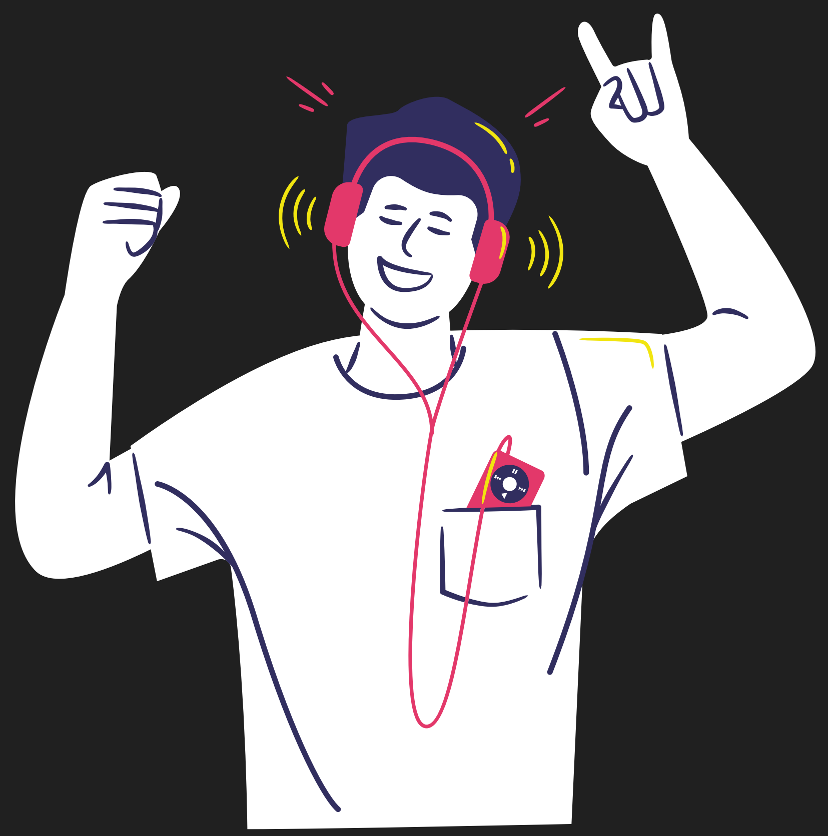


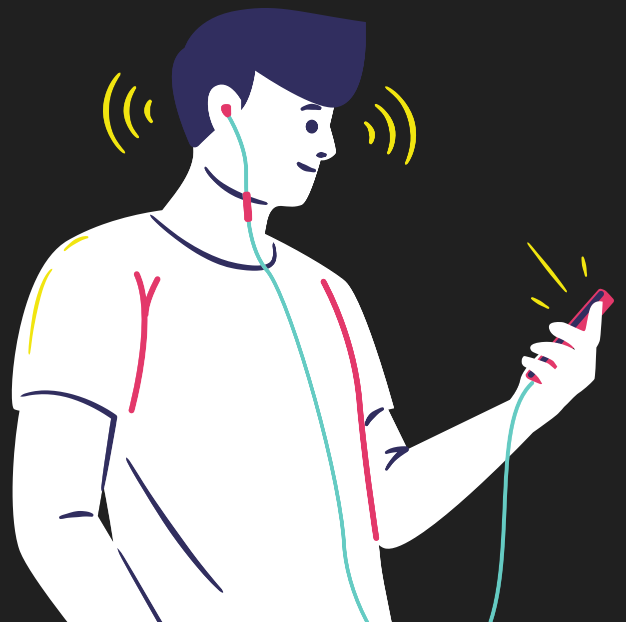
Predicting Song Suggestions Based on Track Metrics

Data 144 Fall 2021

Day 2 Session 1

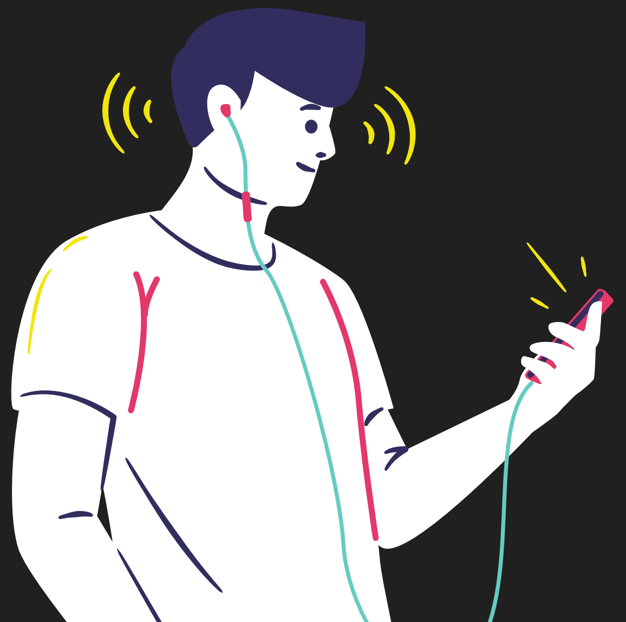
Louie Ortiz, Faizaan Merchant, Aleks Faynleyb, Ryan Sandan

Table of Contents 📚



Introduction

- **Can we accurately predict suggested songs following a playlist, based on the characteristics of a song?**
- Being able to predict songs that have a similar "vibe" can boost creative input, as well as an eclectic music taste
- Ability to "blend" any number of user's songs to recommend tracks that have the highest similarity across all user songs'.

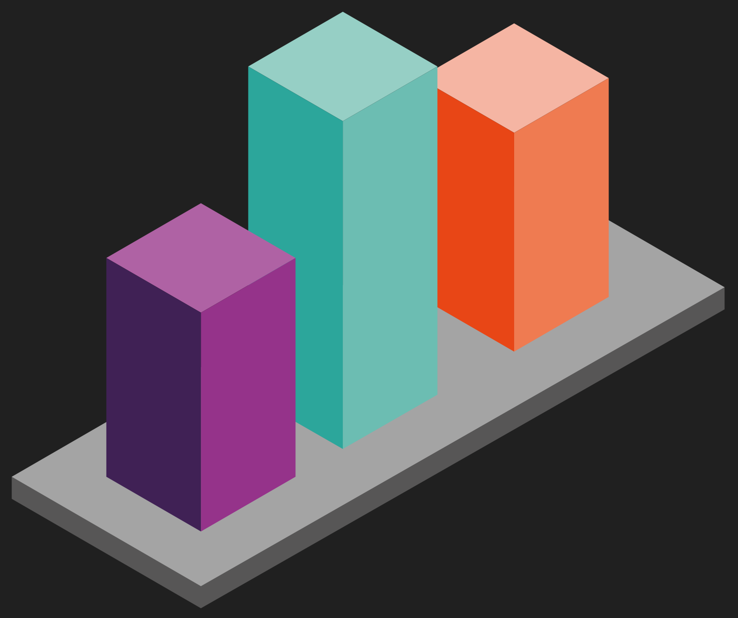


Dataset

- Raw data extracted from Spotify API [database]
- Web API has several endpoints, containing parameters relating to albums, artists, tracks, and playlists
- Includes categorical features like artist and album name, as well as more niche features like "danceability" and "energy"

Dataset

- Since we want to predict song suggestions based on playlist tracks, we meshed playlists together to make one playlist that's large enough to model on
- Created a CSV file consisting of four distinct users, each with their own distinct playlist(s)
- 4539 rows (tracks) x 17 columns (song attributes)

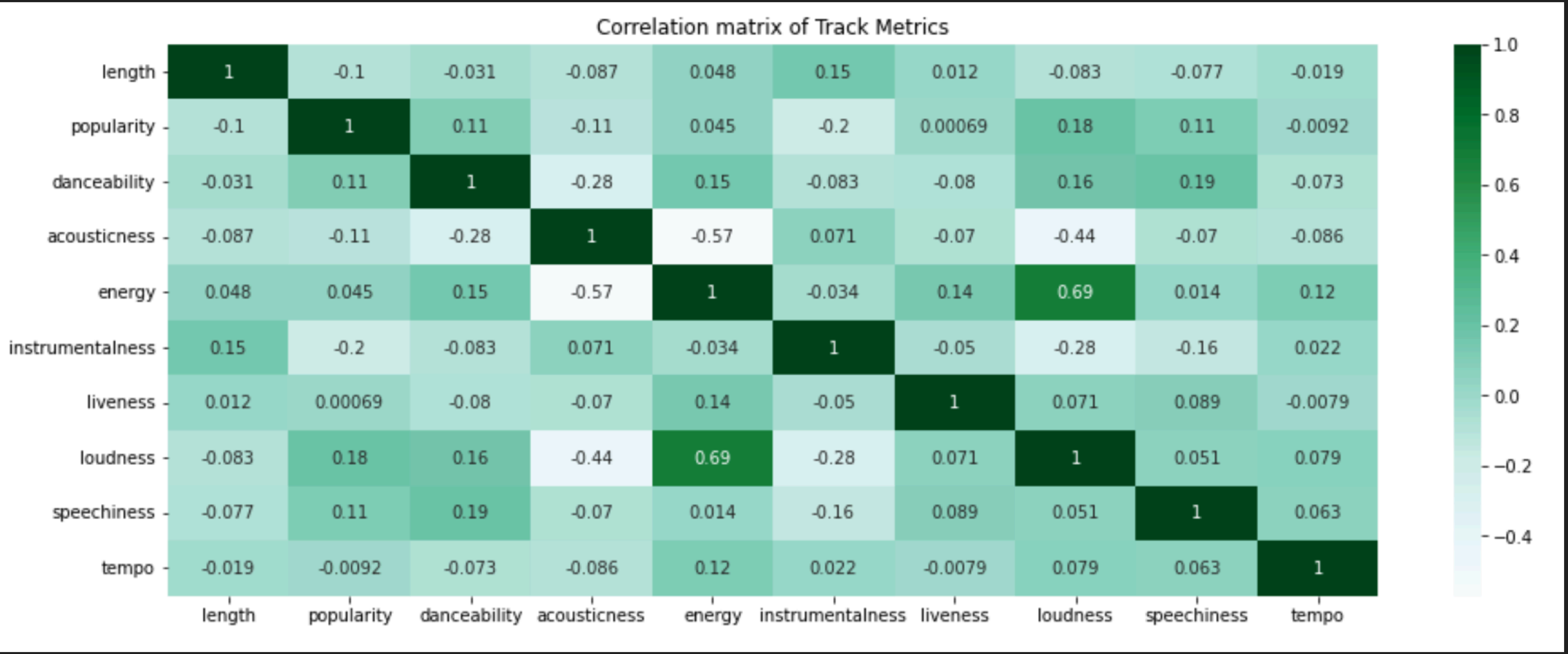


Dataset

- Song data was pulled from three distinct Spotify users, all of whom have different styles of music.

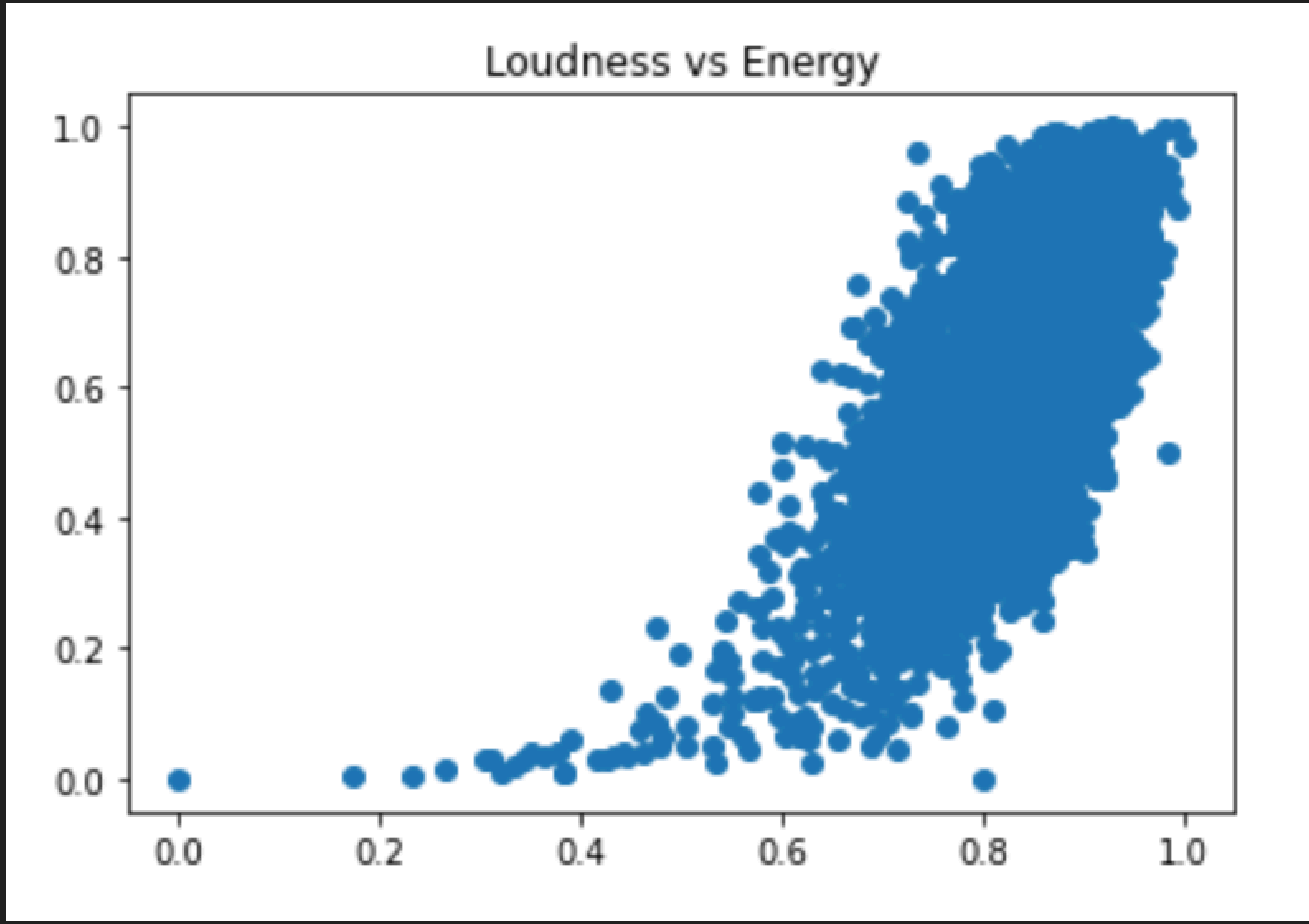
name object	album object	artist object	release_date object	length int64	popularity int64	danceability float64	acousticness floa
Have Yourself a Me… 0.1% Winter Wonderland 0.1% 4377 others 99.8%	R&B: From Doo-Wop … 0.8% Isles Of Wonder: M… 0.8% 3271 others 98.4%	Various Artis… 10.8% Prince 1.7% 1860 others 87.6%	2012-01-01 1.7% 2013-01-01 1.3% 1865 others 97.1%	3056 - 4747301	0 - 98	0.0 - 0.983	0.0 - 0.996
Fight Night	No Label II	Migos	2014-06-03	216247	68	0.874	0.182
Versace (Remix)	Versace (feat. Drake) [Remix] - Single	Migos	2015-01-06	246047	59	0.845	0.0218
Love Songs - Bonus	Parked Car Convos	Kaash Paige	2019-11-15	148640	74	0.641	0.831
Cognac Queen	Tina Snow	Megan Thee Stallion	2018-12-21	222752	74	0.79	0.0593
Wat U Sed (feat. Iamdoechii & Kal Banx)	The House Is Burning	Isaiah Rashad	2021-07-30	176682	64	0.838	0.101

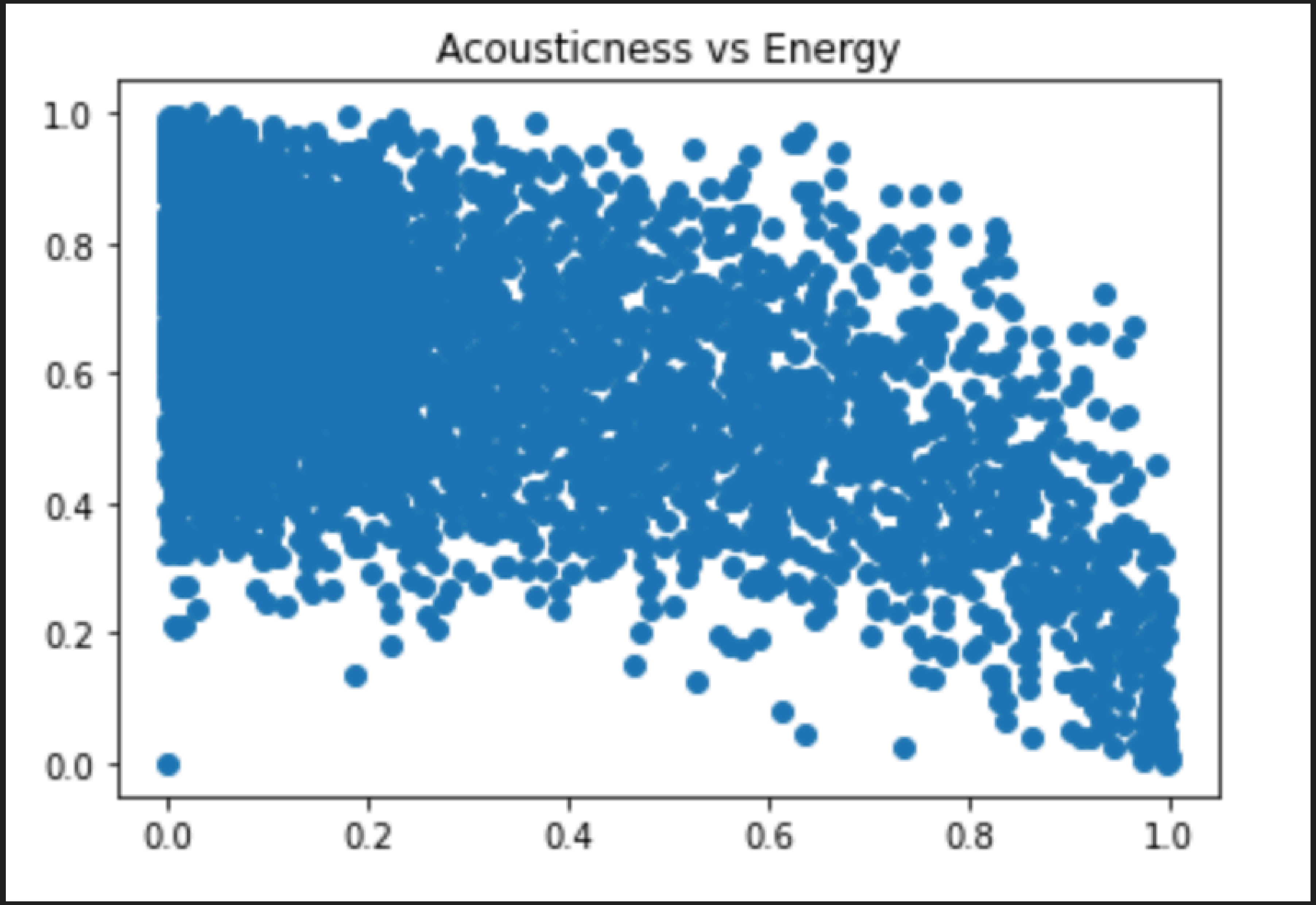
Exploratory Data Analysis



Most notable correlations arebetween the "energy" and "loudness" features (skew positive) and "energy" with "acousticness" (skew negative)

Exploratory Data Analysis

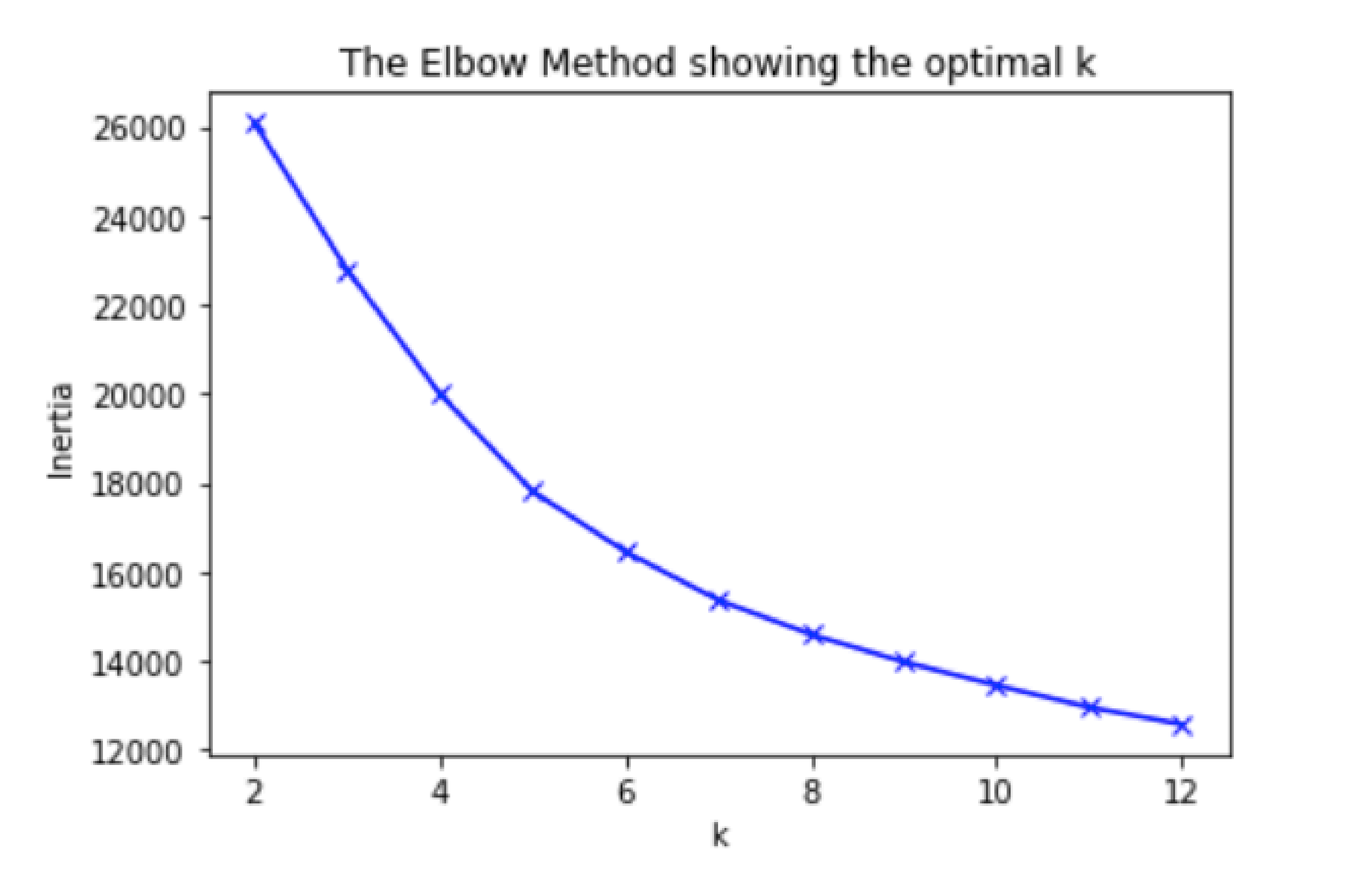




Modeling

Clustering

- To begin our cluster analysis, we used the elbow method to find the optimal number of clusters
- We found that the optimal number of clusters was five
- Given this value, we used sklearn's clustering algorithm on our dataset, creating a table of the cluster labels, dimensional feature values, and track info for each data point or track
- Created a table of cluster centers for use in the recommendation part of our analysis

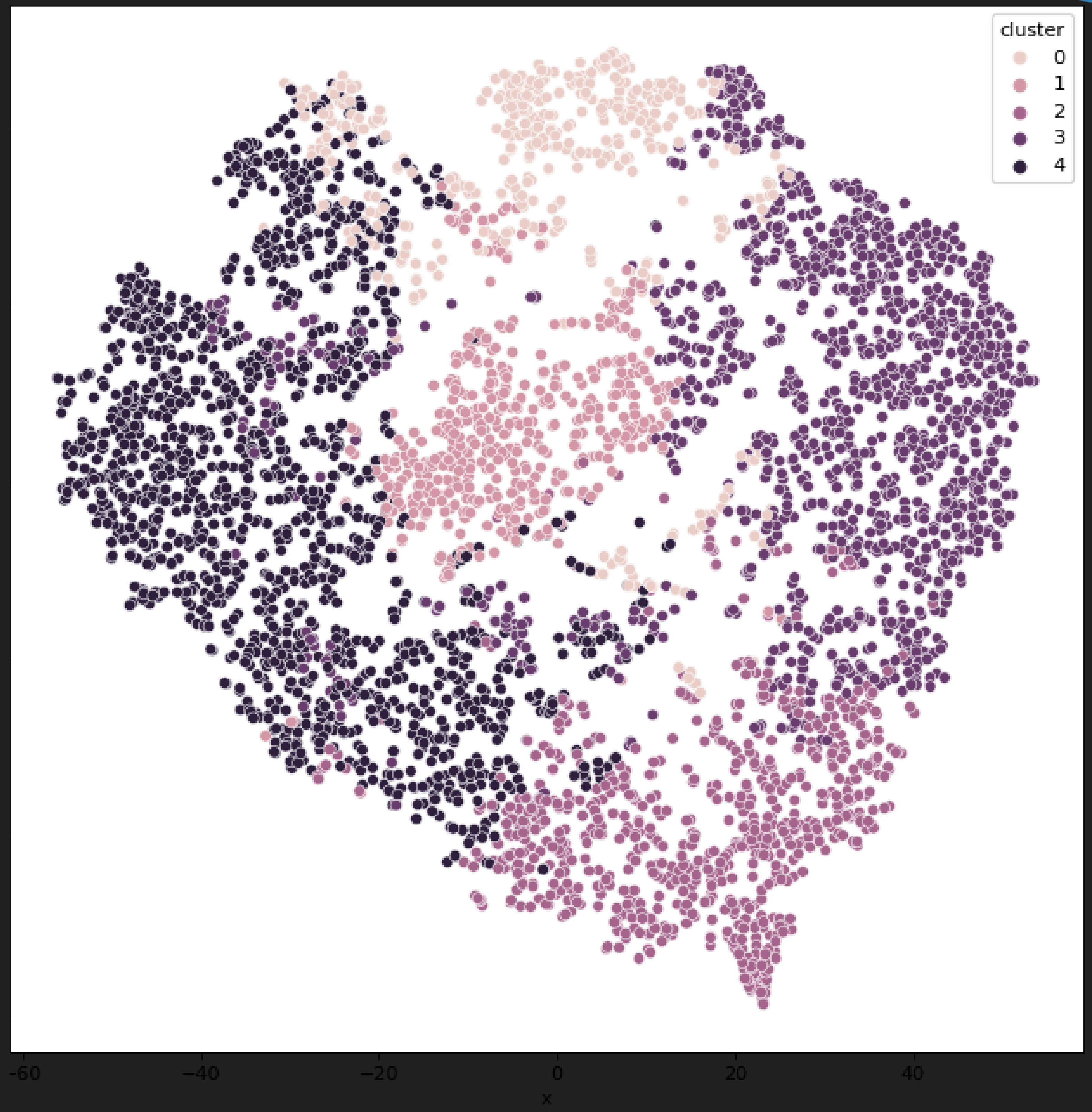


t-SNE

- t-SNE, or t-distributed stochastic neighbor embedding, is a machine learning technique that performs dimensionality reduction on our data
- Given a large number of features, dimensionality reduction helps to downsize this number into a number of components that can easily be graphed
- We used t-SNE to reduce the dimensions of our data point table and cluster center table
- This allowed us to better visualize, analyze, and understand how each data point fit into a given cluster

Clustering

- Performed t-SNE embedding on our dimension reduced data to give us five clusters (from optimal k-mean cluster)



Random Track Subsetting

- We needed to randomly select a large dataset of new songs from Spotify's API
- Created a function which called on Spotify's search endpoint, adding a random letter to the final query string
- Ue used a wildcard character (%), randomly entering it after or on both sides of the random letter within the final query string (e.g. 'd%' or '%d%')
    - Allowed us to search for any song that matched the random letter that was entered
- Randomly selected the offset parameter, which specified where in the returned data, to start picking songs from
- Ended with a new dataset of 1,000+ randomly selected songs

Recommendation Analysis

```python
column_names = ["name", "album", "artist", "distance"]
# test_centers_trimmed = test_centers.iloc[:,4:]

# calculate the Euclidean distance between two vectors
def euclidean_distance(row1, row2):
    distance = 0.0
    for i in range(len(row1)-1):
        distance += (row1[i] - row2[i])**2
    return np.sqrt(distance)

#finds the distance to the closest cluster for a given song
def find_distance_from_centers(song, centers):
    distances = []
    for i in range(len(centers)):
        distance = euclidean_distance(song, centers.iloc[i, :])
        distances.append(distance)
    return min(distances)

#creates a dataframe that includes a best distant column
def near_cluster_centers(centers, songs):
    songs_distances = pd.DataFrame(columns = column_names)
    for i in range(len(songs)):
        song_trimmed = songs.iloc[i,5:12]
        best_distance = find_distance_from_centers(song_trimmed, centers)

        songs_distances.loc[i, "name"] = songs.iloc[i,0]
        songs_distances.loc[i, "album"] = songs.iloc[i,1]
        songs_distances.loc[i, "artist"] = songs.iloc[i,2]
        songs_distances.loc[i, "distance"] = best_distance

    return songs_distances

distance_df = near_cluster_centers(cluster_center_table, random_songs_df)
distance_df.sort_values("distance")
```

- After gathering random Spotify tracks, we needed to write a function that measured the distance to the closest cluster for any given song
- Songs closest to our 5 cluster centroids had the highest similarity across all of our metrics
- The songs with the smallest distance were then recommended

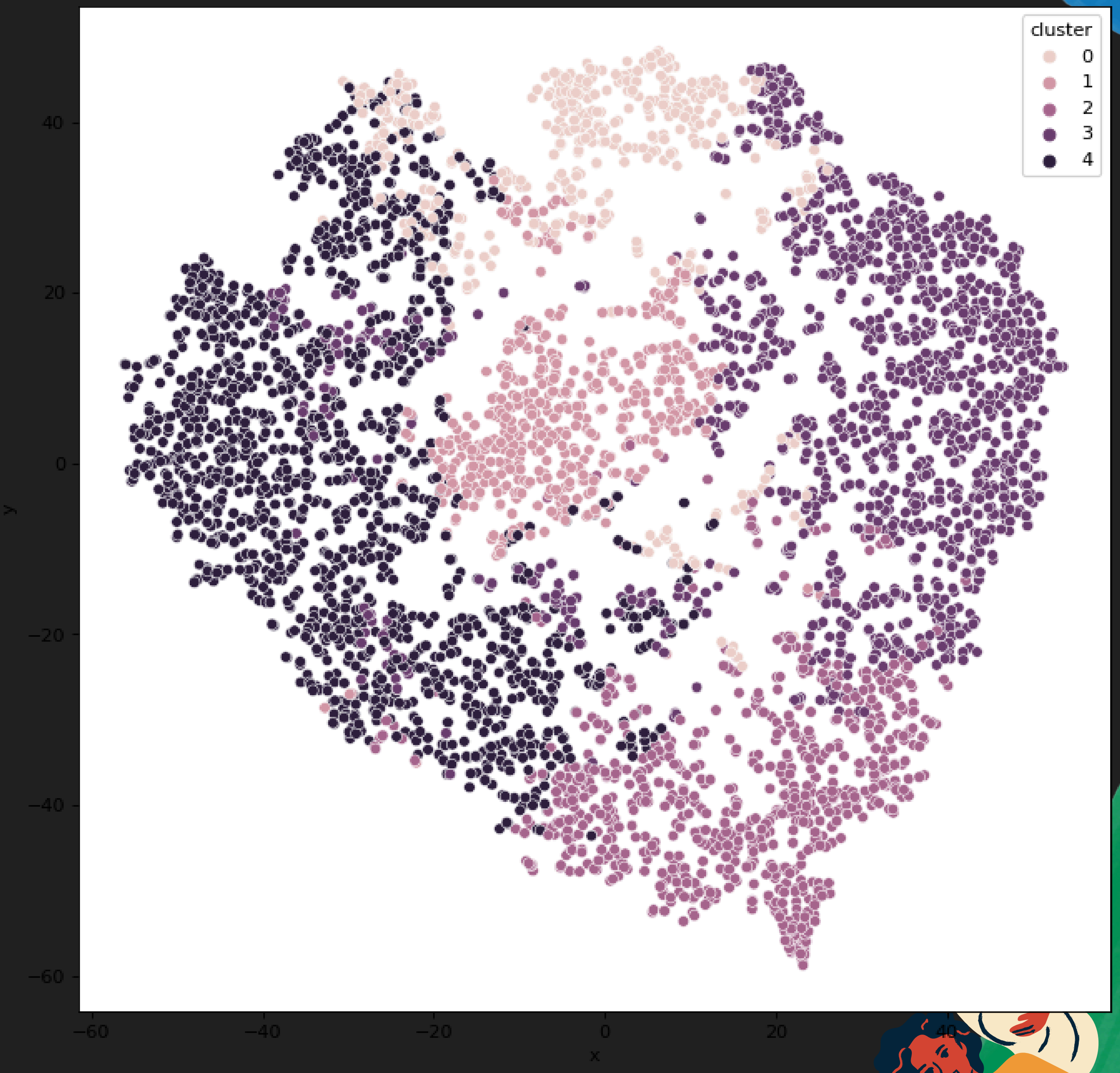
Results

- Suggested songs relative to cluster centroids based on metrics

	name object Quizás Si, Quizá… 0.2% Kings & Queens 0.2% 1052 others 99.6%	album object Red (Taylor's Vers… 1.9% Fighting Demons 1.5% 844 others 96.6%	artist object Taylor Sw… 3% Juice WRLD 3% 572 others 94%	distance object 0.7668313278239… 0.1% 1.170928228414З… 0.1% 1061 others 99.8%
351	Balance ton quoi	French Song's For New Year's Eve	Angèle	0.7668313278239661
615	The Christmas Song (Chestnuts Roasting On An Open Fire)	Christmas for Kids	Justin Bieber	1.1709282284143183
435	Lover	Music Superstars	Taylor Swift	1.2155788691666856
282	Ready	Trending Now Volume 39	Lil Baby	1.2416300444302046
356	Quihubo Cuando - En Vivo	Quihubo Cuando (En Vivo)	Carin Leon	1.281367592575012

Conclusion

- Although we modeled and analyzed a dataset with over four thousand features (tracks), more expansive data would be useful in producing more accurate predictions
- Standardizing data would help in making sense of our analyses
- Not enough memory to do more training/modeling
- Learning how to suggest more "similar" tracks is useful for user experience and stakeholder (Spotify's) profit.



Thank you!