Song Recommendation System

Son Nguyen, DSI-111

Problem Statement

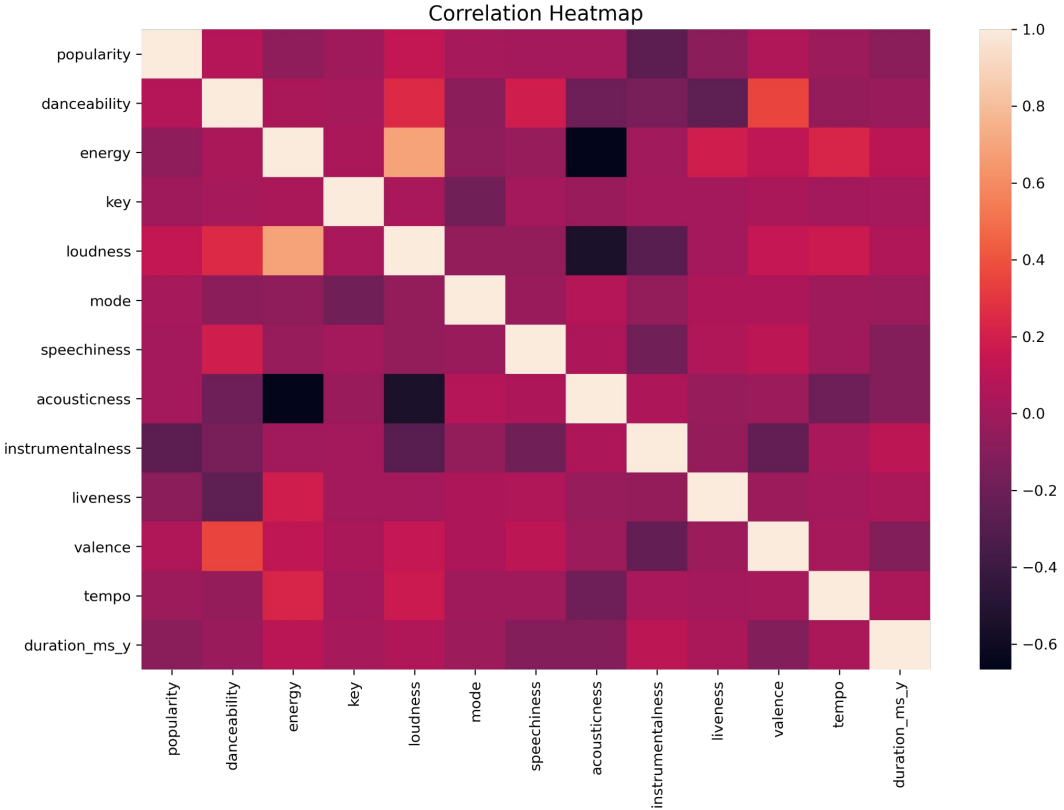
- Music is considered a “Cultural Universal”, it is often related to the origin of language
- There are dozens of ‘base’ music genres and thousands of subgenres
- Spotify has 82 million tracks on their platform.
- Have you ever asked yourself: ‘I wish I could find more songs like xyz’?
- Can we build a ‘Cold Start’ recommender system for personal use cases
- Examples: DJ may want to find similar songs for smooth transitions and audience engagement

Background

- Using Spotify's API, 82000+ song and features were pulled from artists albums
- Searched by 'Artist' from list of artists
- Pulls are far from perfect, JSON indices change from artist to artists.

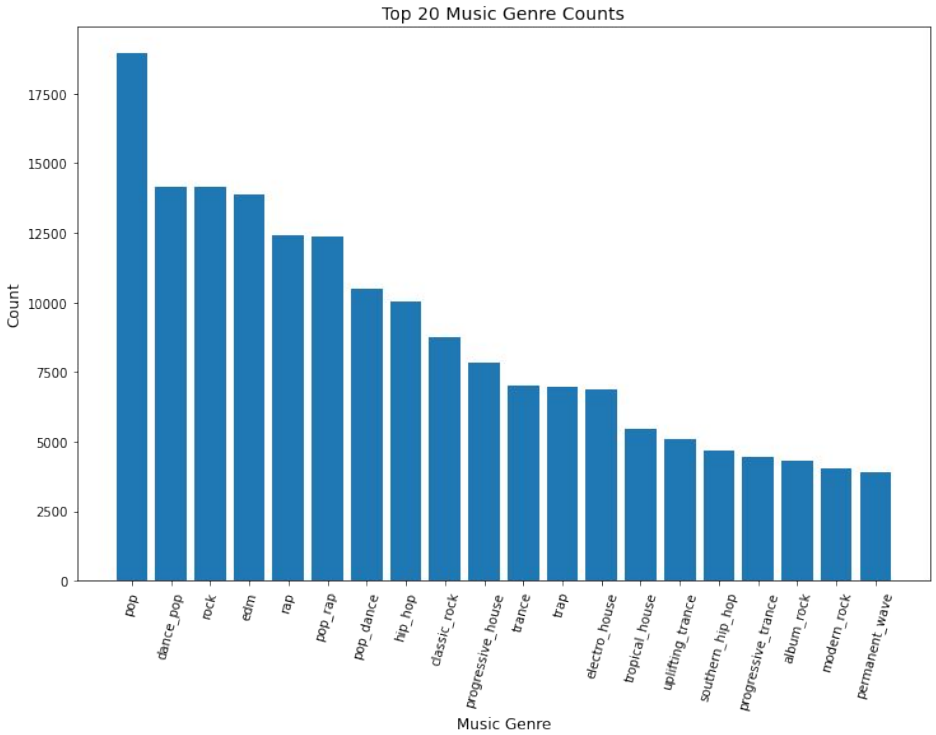
Feature	Type	Description
Tempo	Float	Speed/Pace
Danceability	Float	Danceability
Energy	Float	Sound Energy
Key	Int	note/chord
Speechiness	Float	verbal detection
Loudness	Float	Loudness Unit Full Scale
Acousticness	Float	How acoustic song is
Instrumentalness	Float	How instrumental song is
Liveness	Float	Detects live crowd
Valence	float	Mood of song
Popularity	Int	How popular track is

Exploratory Data Analysis

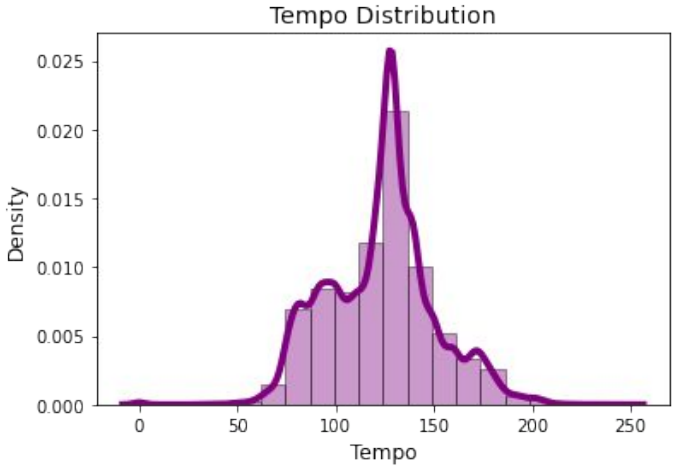


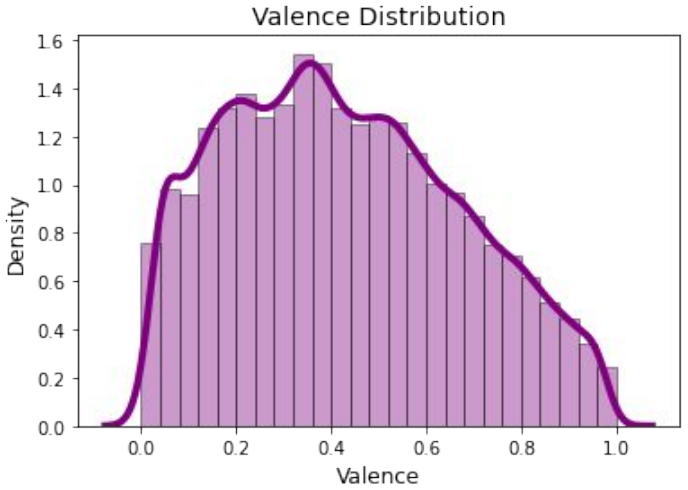
Exploratory Data Analysis

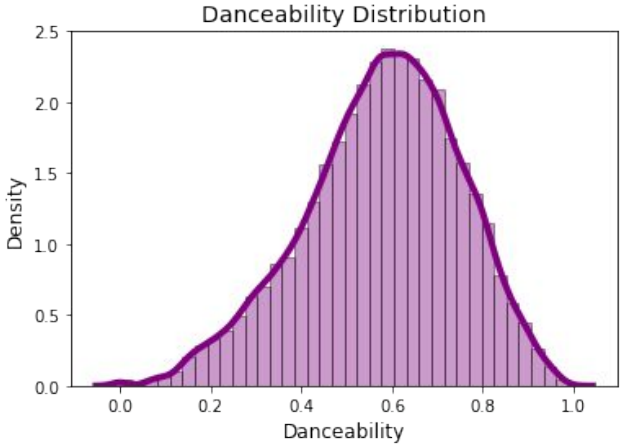
- ‘Pop and Rock’ showed up in 7/20 in Top Genres
- EDM and EDM sub genres appeared in 9/20
- 1230 Total genres from pulls

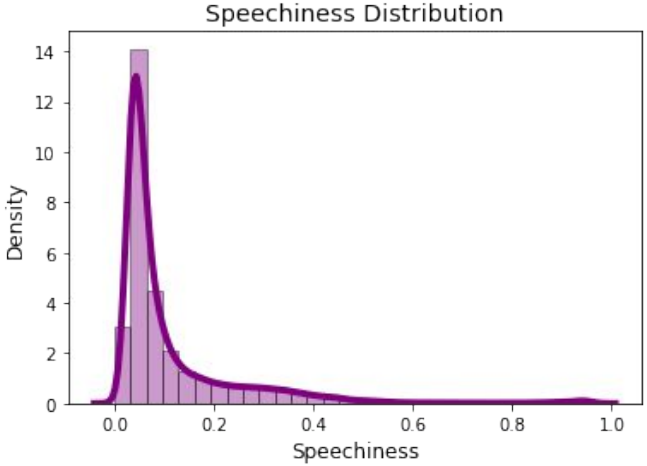


Exploratory Data Analysis



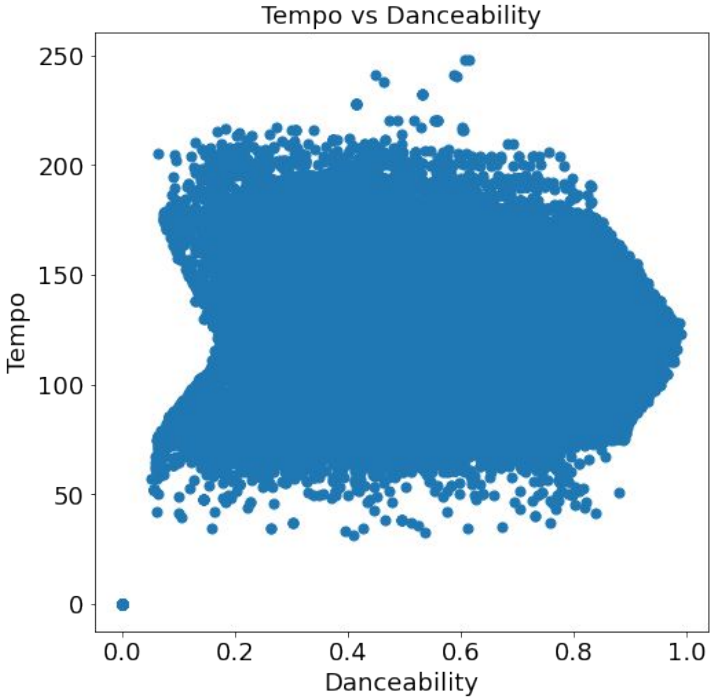


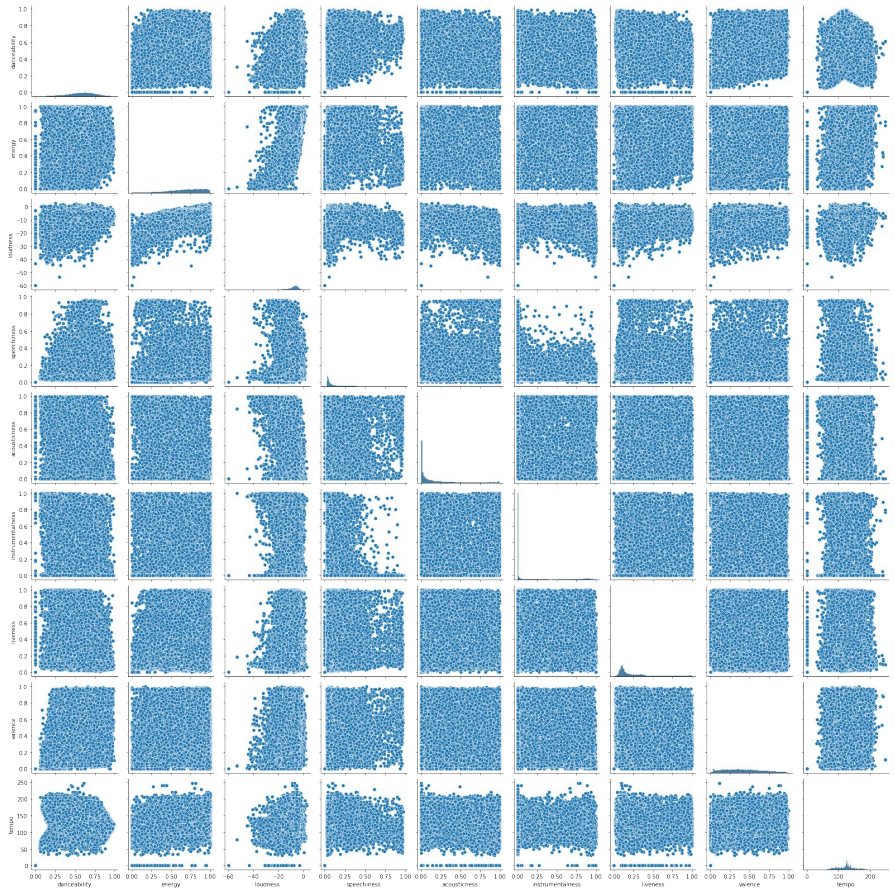




Exploratory Data Analysis

- Tempo appears to have some impact on the danceability of a track





Tackling Cold Start Problem

- **Cosine Similarity** using features.
- Created ≈ 6.7 million calculations
- Represented in matrix

Song	song_1	song_2	song_3	song_4	song_5
song_1	1.0	0.6	0.8	0.4	0.0
song_2	0.6	1.0	0.5	0.9	0.2
song_3	0.8	.5	1.0	0.7	0.7
song_4	0.4	0.9	0.7	1.0	0.3
song_5	0.0	0.2	0.7	0.3	1.0

Pros

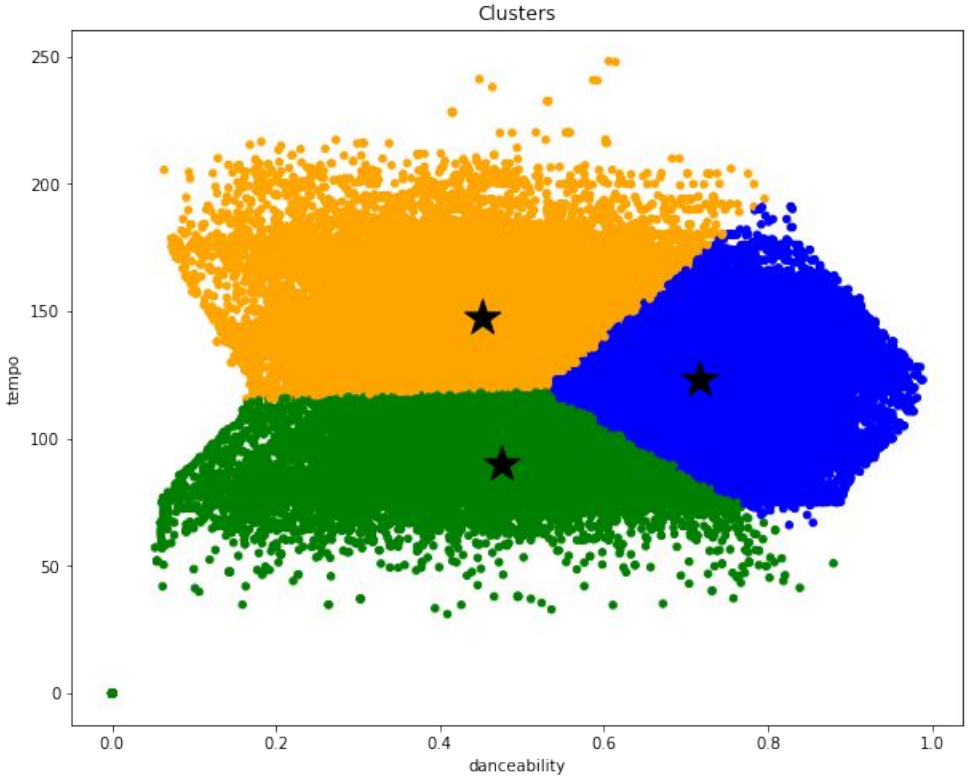
- Using TF IDF got great predictions on same artist and similar artists

Cons

- Biased for genre name
- Matrix size
- Little variability in songs

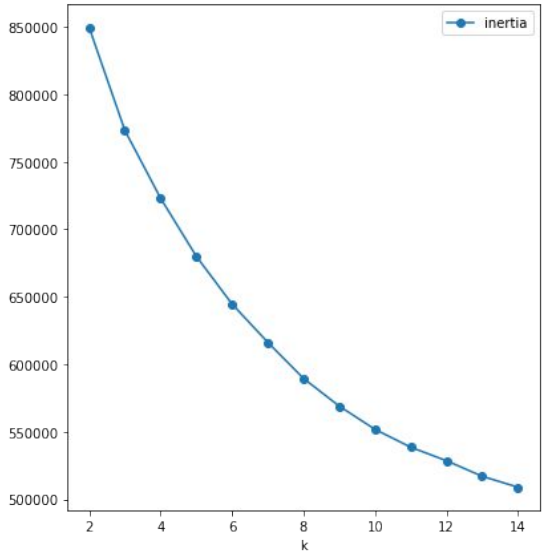
K - Means

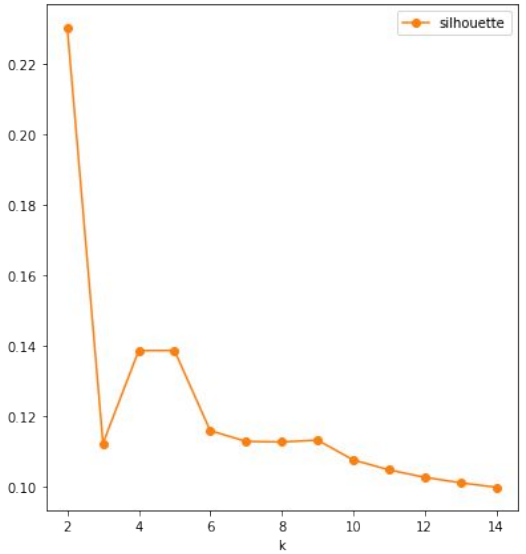
	k	inertia	silhouette
0	2	109492.290133	0.336848
1	3	70854.650002	0.371101
2	4	56749.082924	0.343542
3	5	44826.043254	0.359776
4	6	37506.505362	0.364979
5	7	32448.079873	0.360274
6	8	28093.525463	0.359454
7	9	25081.004793	0.347796



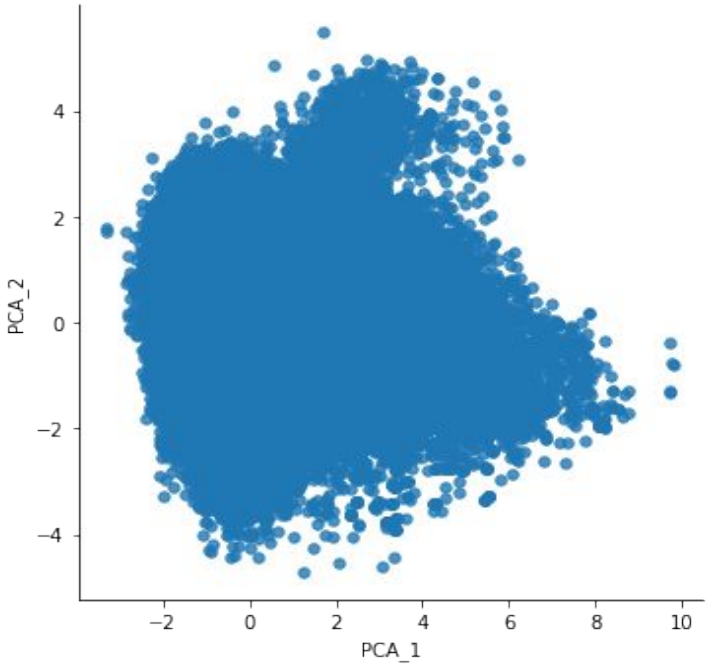
K - Means (All Features)

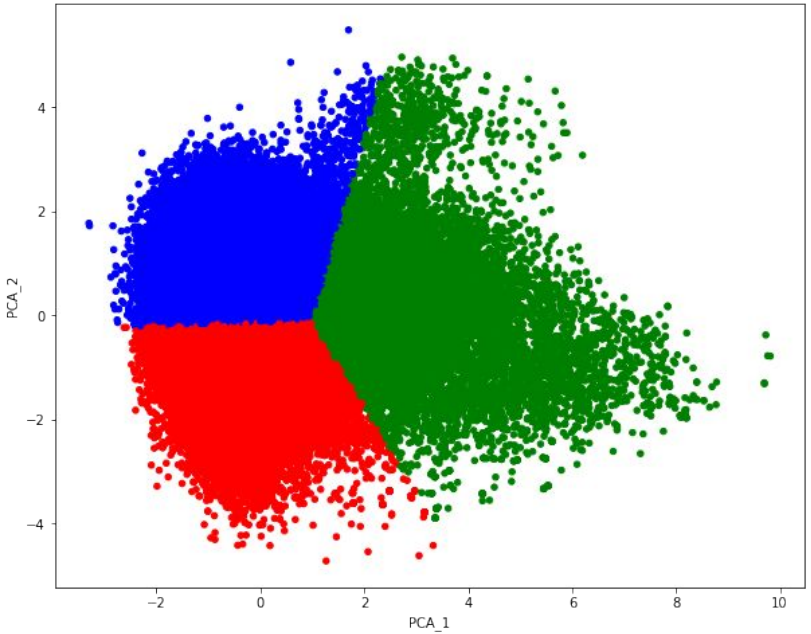
	k	inertia	silhouette
0	2	849443.738087	0.230377
1	3	773125.619015	0.112486
2	4	722902.131665	0.138722
3	5	679857.433799	0.138674
4	6	644514.529443	0.116022
5	7	615639.614105	0.118374
6	8	589423.760903	0.112822
7	9	569533.163362	0.111897



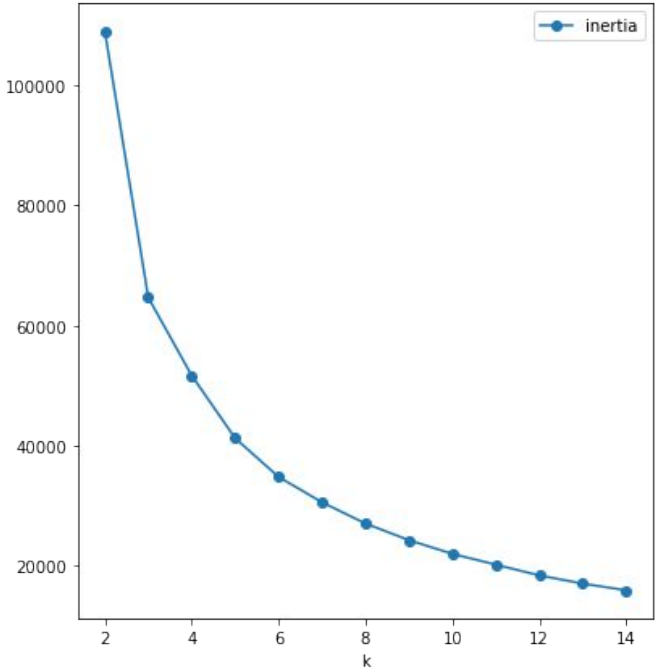


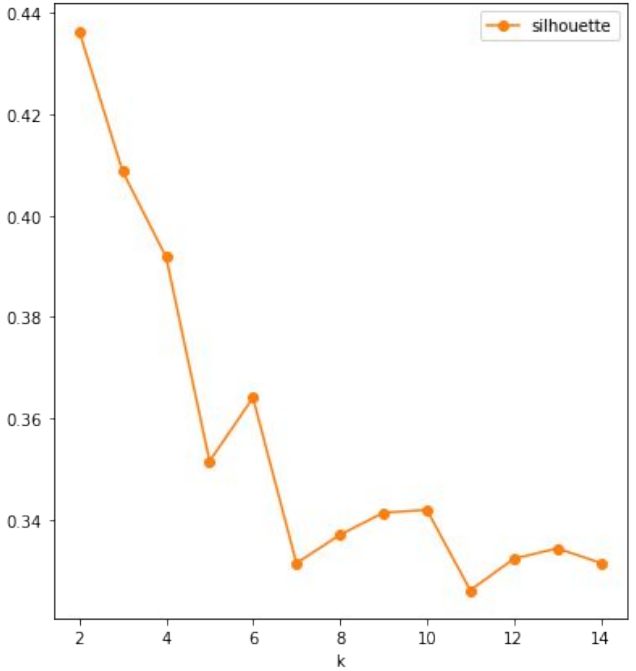
PCA & K-Means





PCA & K-Means


inertia
100000
80000
60000
40000
20000
2
4
6
8
10
12
14
k


silhouette
0.44
0.42
0.40
0.38
0.36
0.34
2
4
6
8
10
12
14
k

Recommendations

- If we are quickly looking for similar artists and albums we can use the cosine similarities matrix for quick findings
- K-Means will give a good variety of recommendations
- PCA should be implemented with K-means if we vectorize genres with tfidf

- More research and data is needed for optimizing the models.