Spotify Recommender.

By, Robert Huey

Table of contents

Introduction

I wanted to build a music recommender for spotify based on metrics used in music and could be found via the API.

Gathering data.

Originally I was using the API, but it wouldn't connect to the live host to gather recommendation data. So I found this lovely application that would allow me to strip the info from spotify by using the URI paths to playlists. I then created a bunch of playlist URI's and ran them through the application to gather my music data and turn them into csv's.

By, plamere
Website -
http://organizeyourmusic.playlistmachinery
.com/

The Track Properties

Descriptions by, plamere.

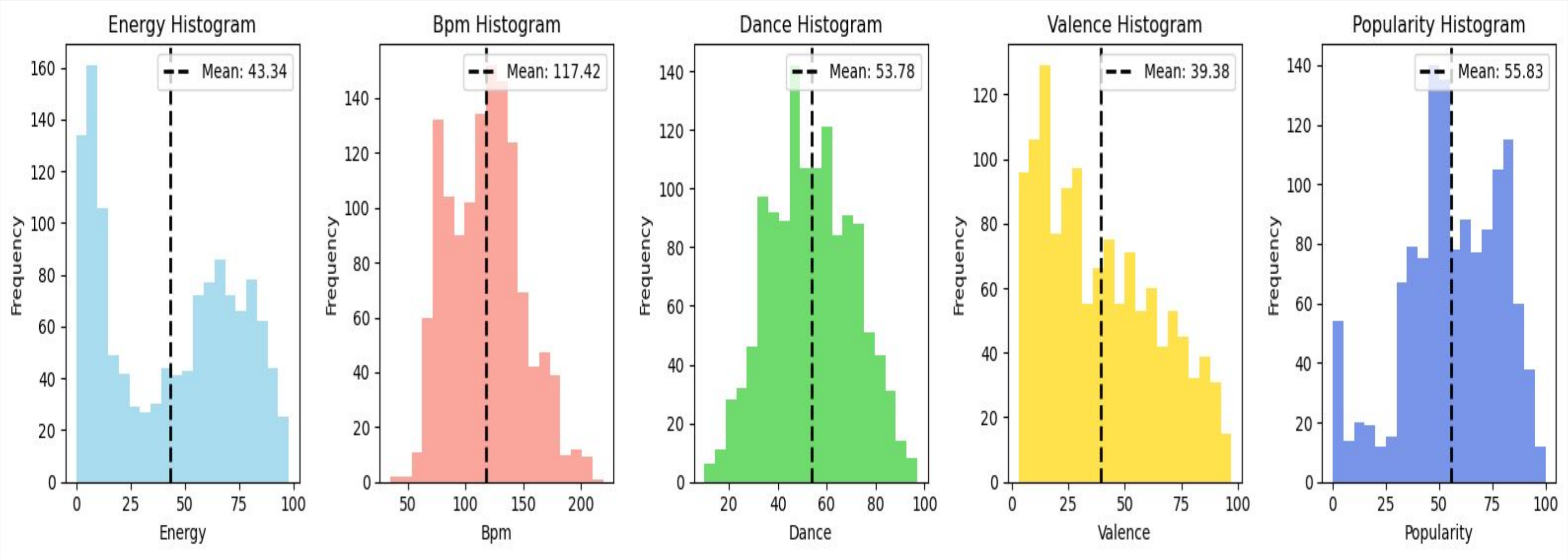
1. **Genre** - the genre of the track
2. **Year** - the release year of the recording. Note that due to vagaries of releases, re-releases, re-issues and general madness, sometimes the release years are not what you'd expect.
3. **Added** - the earliest date you added the track to your collection.
4. **Beats Per Minute (BPM)** - The tempo of the song.
5. **Energy** - The energy of a song - the higher the value, the more energtic. song
6. **Danceability** - The higher the value, the easier it is to dance to this song.
7. **Loudness (dB)** - The higher the value, the louder the song.
8. **Liveness** - The higher the value, the more likely the song is a live recording.
9. **Valence** - The higher the value, the more positive mood for the song.
10. **Length** - The duration of the song.
11. **Acousticness** - The higher the value the more acoustic the song is.
12. **Speechiness** - The higher the value the more spoken word the song contains.
13. **Popularity** - The higher the value the more popular the song is.
14. **Duration** - The length of the song.

Cleaning.

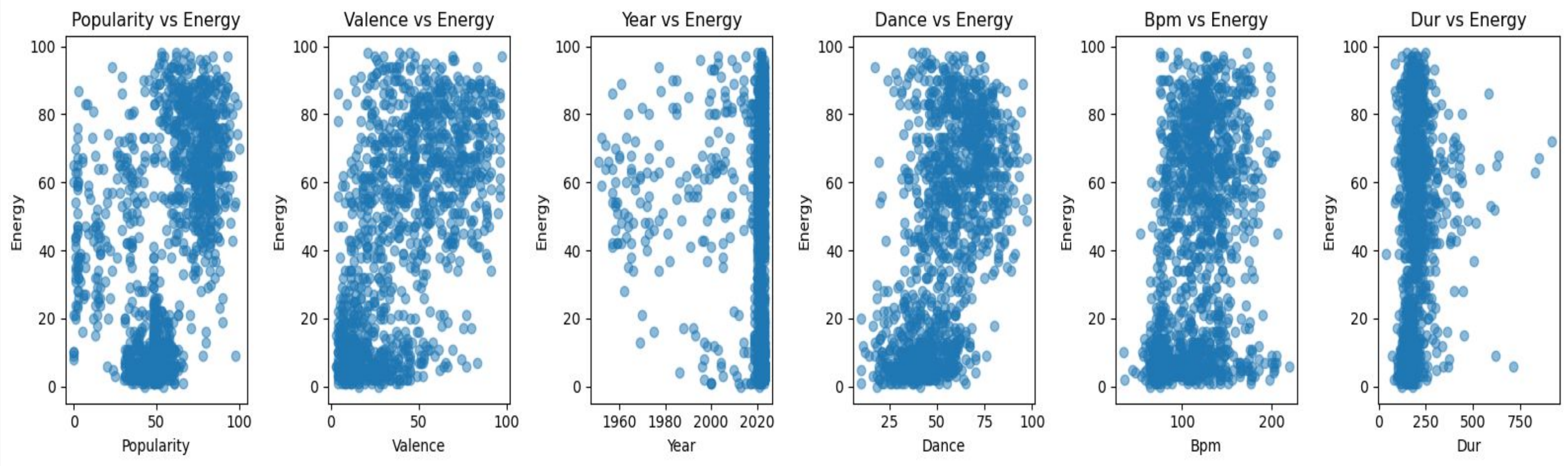
Artists that belonged to multiple genre's would show up as NaN values. So I assigned them the 'alternative' genre to not lose a large amount of data. I also re-named the columns and deleted duplicate songs.

Exploratory data analysis.

First, I wanted to take a look at the feature distributions separately to see what insights I could pull from the graphs and the playlists I concatenated.



Then I wanted to see if the data clustered anywhere.

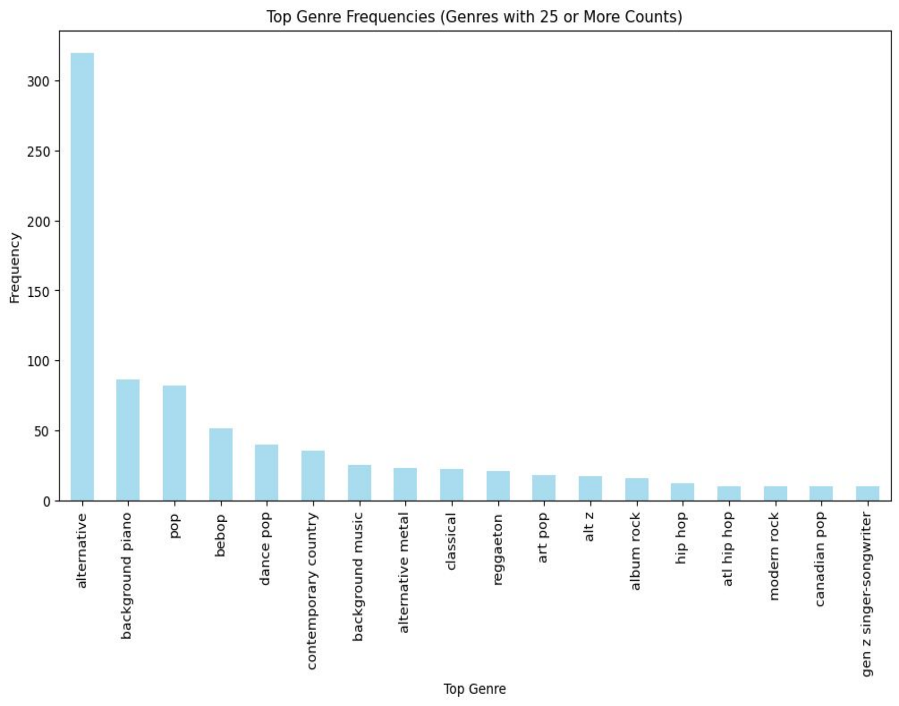


Popularity had about 3 clusters and valencer with 2.

Year was just 1. Danceability had 2 clusters.

BPM semi has 2 and duration is just 1 cluster.

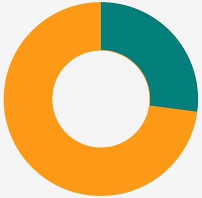
Lastly, I wanted to see how the top 25 genres were distributed.



Modeling.

Looking at coefficients for energy.

28%



Popularity

Holding energy constant, for every 1 unit increase in popularity we see a .28 increase in energy.

40%

Valence

Holding energy constant, for every 1 unit increase in valence, we will see a .40 increase in energy.

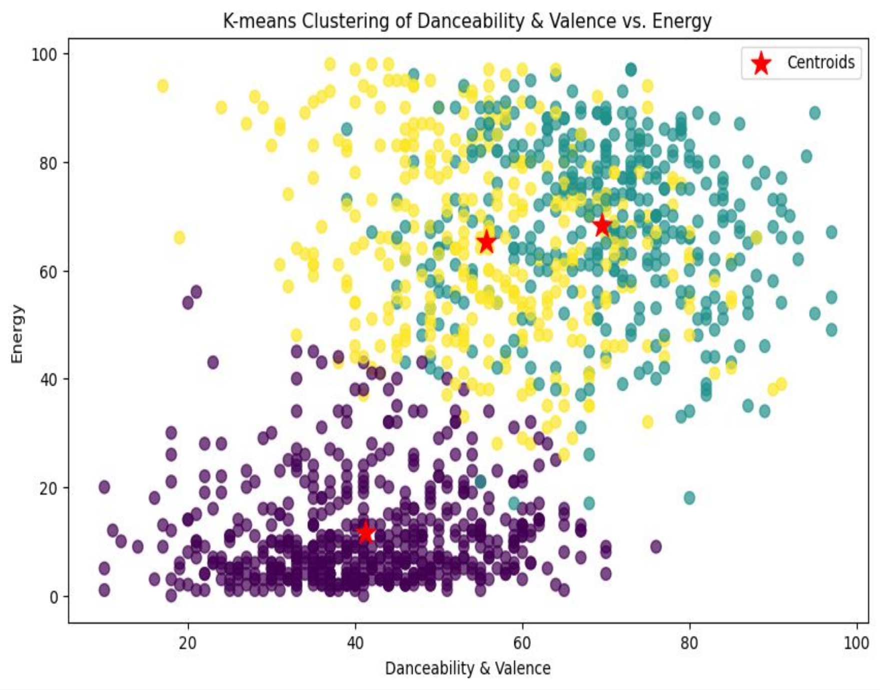
50%

Danceability

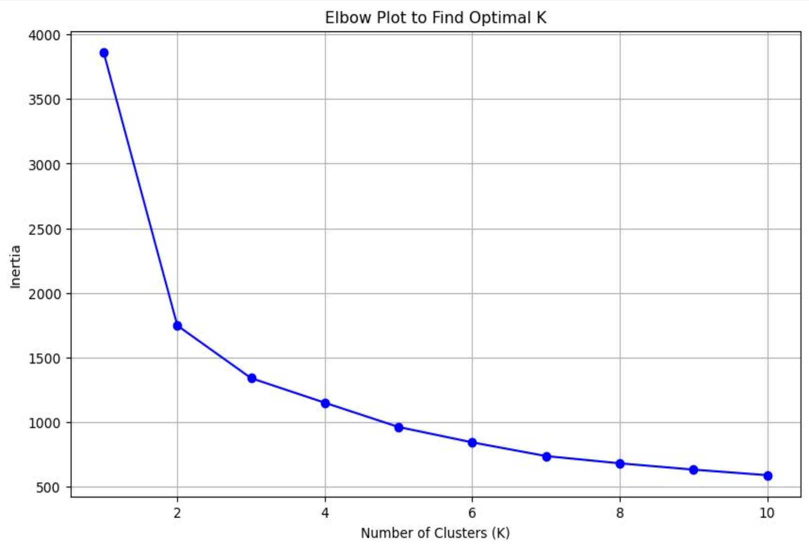
Holding energy constant, for every 1 unit increase in danceability, we will see a .50 increase in energy.

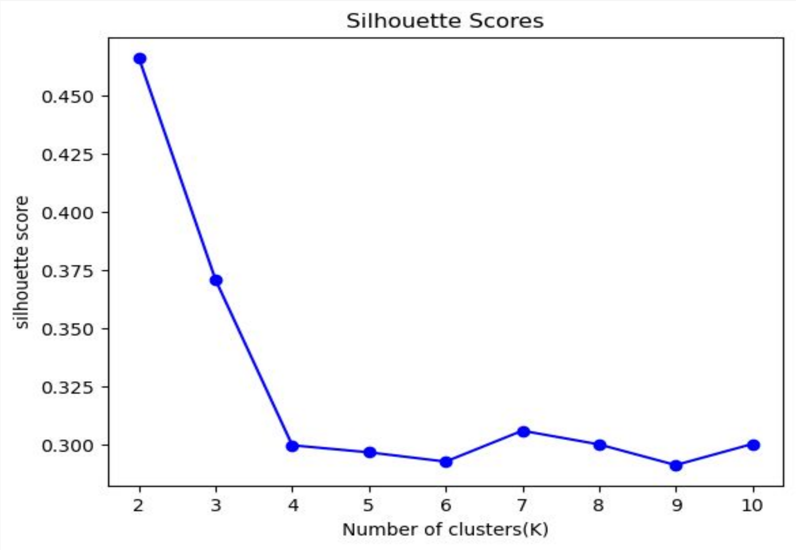
K-means Clustering.

I created a k-means clustering model with the top 2 features that correlated well with energy. Setting the value of K to 3 to see if it will vary after finding the correct value of k via inertia and silhouette scores.



We can see that both elbow plots prefer 2 clusters.





DBSCAN

2 clusters.

Was the best it could do at epsilon = .39

Silhouette Score

Was a .14...

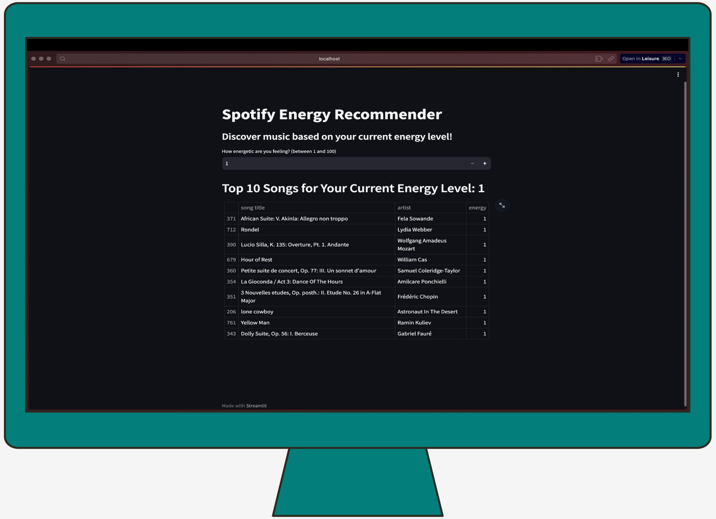
Cluster sizes.

0 = .95
-1 = .05

Biggest differences.

Acoustic = 20 point difference.
Energy = 13 point difference.

Streamlit.



Let's have some fun.

Conclusion.

While the recommender system actually worked well in in assigning music based off current energy levels it would probably be better if I took out the top billboard hits and focused on moods more. Primarily, because I want clearer clusters to then more accurately recommend based off energy and mood.

Thanks!

Do you have any questions?
roberthuey94@gmail.com