Chandler Ebrahimi

Final Report

Capstone 3

Recommendation System for Spotify Music Dataset

Problem Statement:

For people who listen to music and want to find new songs. People want to find new songs that are similar to a song they already know; a recommendation system can be used to find songs similar to a song they already know. I will be using a dataset that contains songs and genres were extracted from an Spotify api into the dataset, that was provided on kaggle as CCO: public domain, free of use.

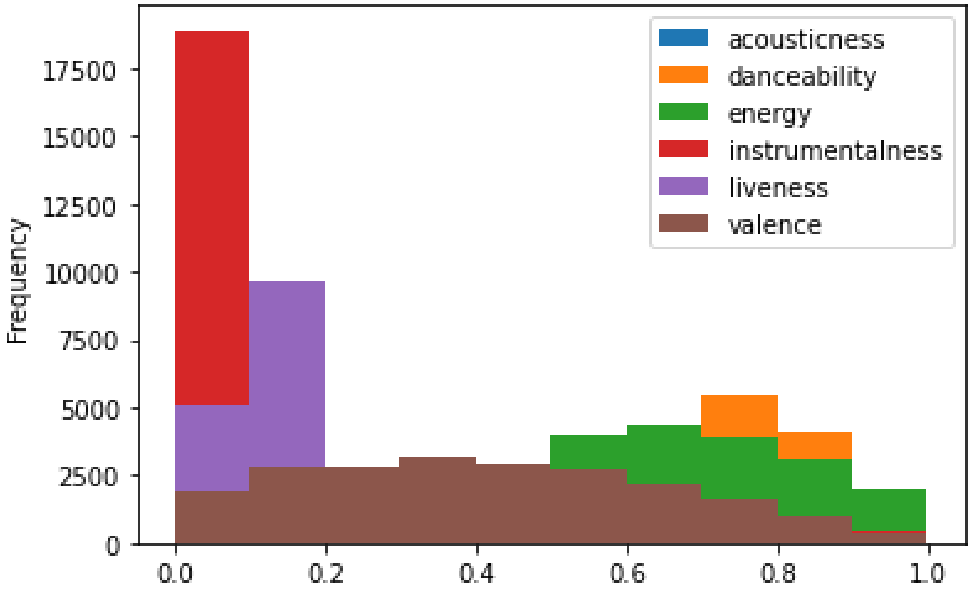
Data Wrangling:

The original dataset has 42305 rows with 22 columns. The useful features on the dataset are the *sound features* of the song such as 'danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness’, ‘acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo', 'duration_ms', 'time_signature'. Where I was able to create a recommendation system out of these features from the dataset.

Some columns of data were irrelevant to the recommendation system and other columns were made up of at least 55+% of null or missing values. These columns were the following: 'Unnamed: 0', 'title', 'analysis_url', 'track_href', 'id', 'uri', 'type'.

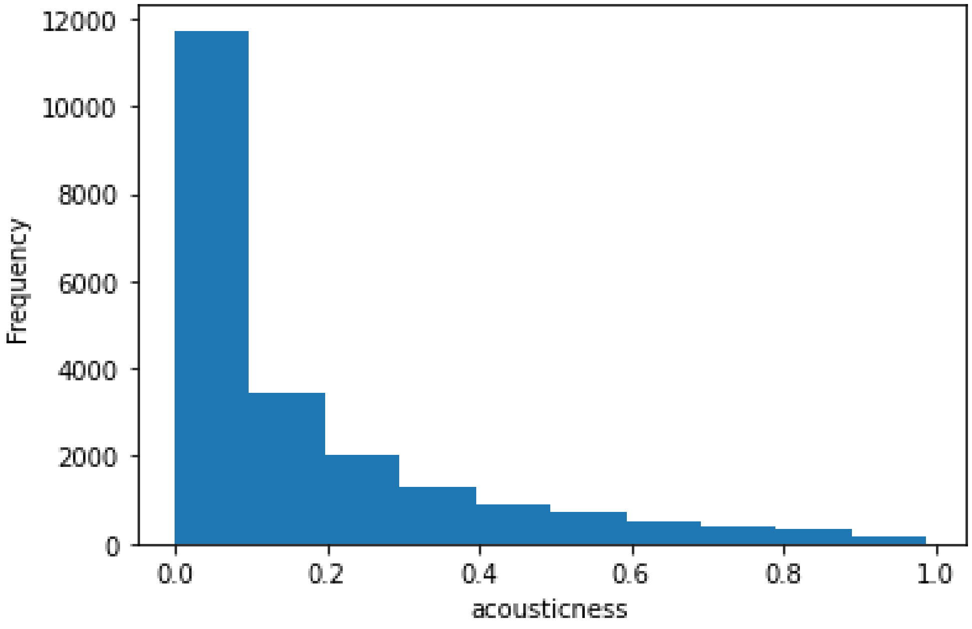
Exploratory Data Analysis:

For Exploratory Data Analysis, I choose what I think are the more prominent sound features out of the list all the sound features, 'acousticness', 'danceability', 'energy', 'instrumentalness', 'liveness', 'valence'. A graph shows the overall precent value of where the songs lies.

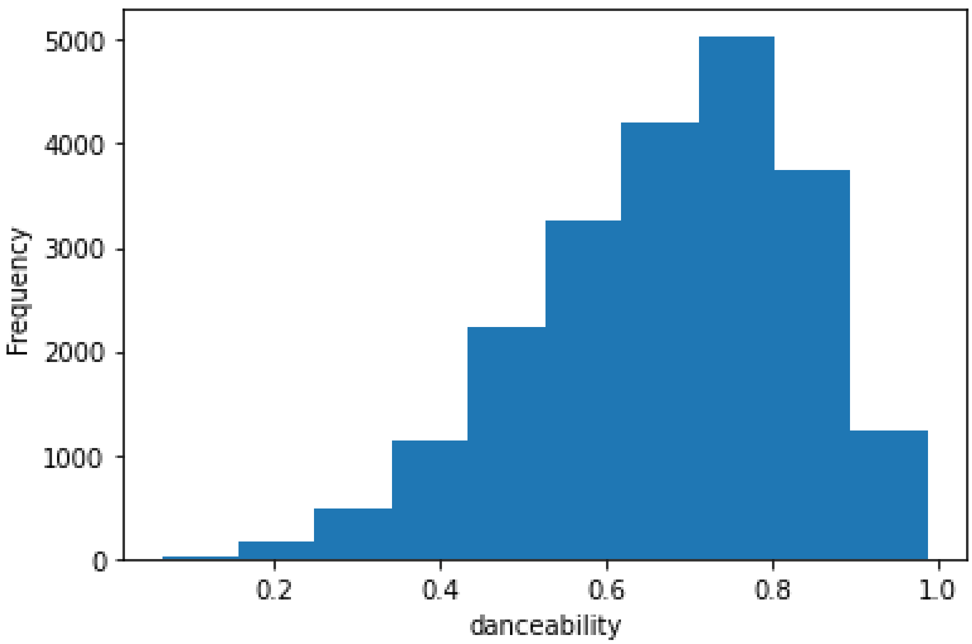


We can see that most songs have less than 15 percent of the song being instrumental, while energy level of the song is on the higher end, typically people today like music with high energy and high danceability. However, when graphing all on one plot 'acousticness' and other parts of other features are being covered up.

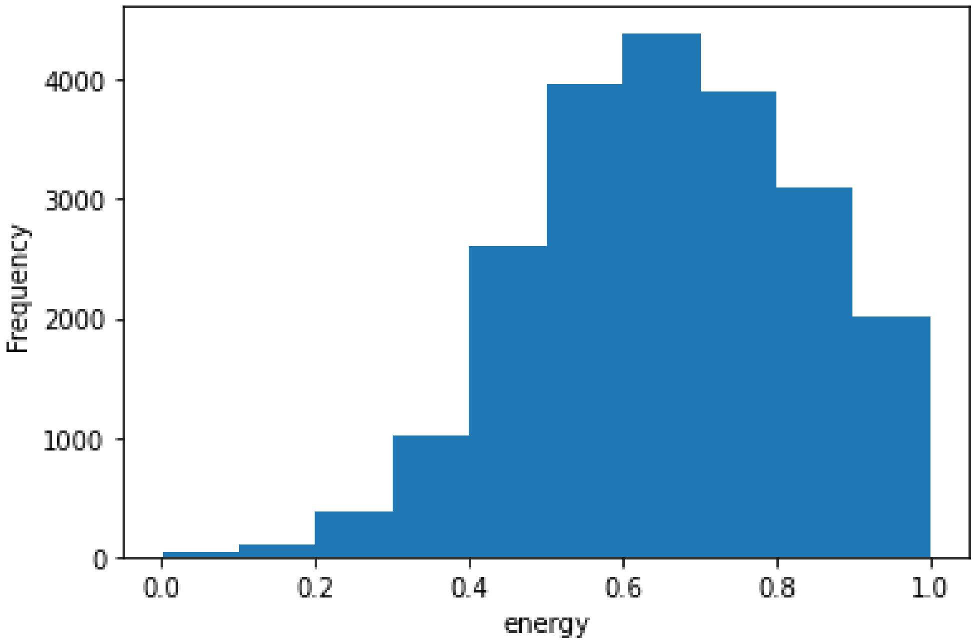
Breaking each feature into its separate graph we see the following:



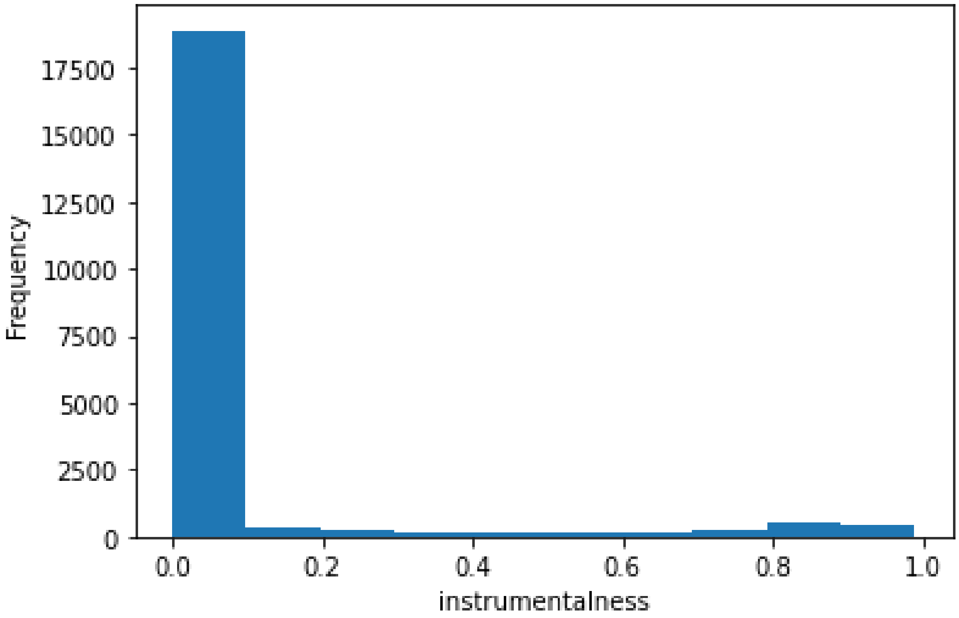
'Acousticness' is generally on the lower side of songs



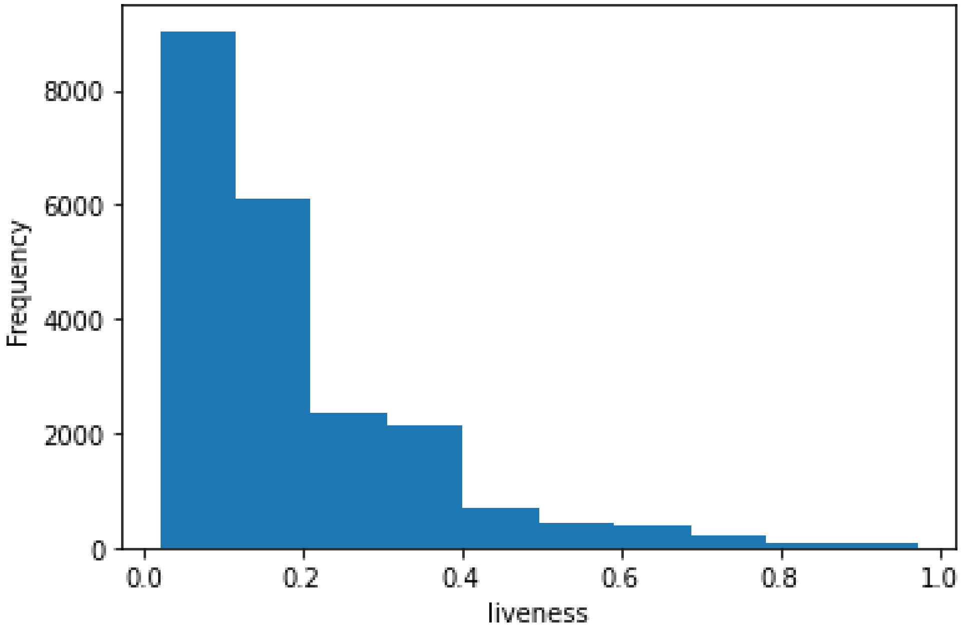
Songs typically have a higher ‘danceability’ to them.



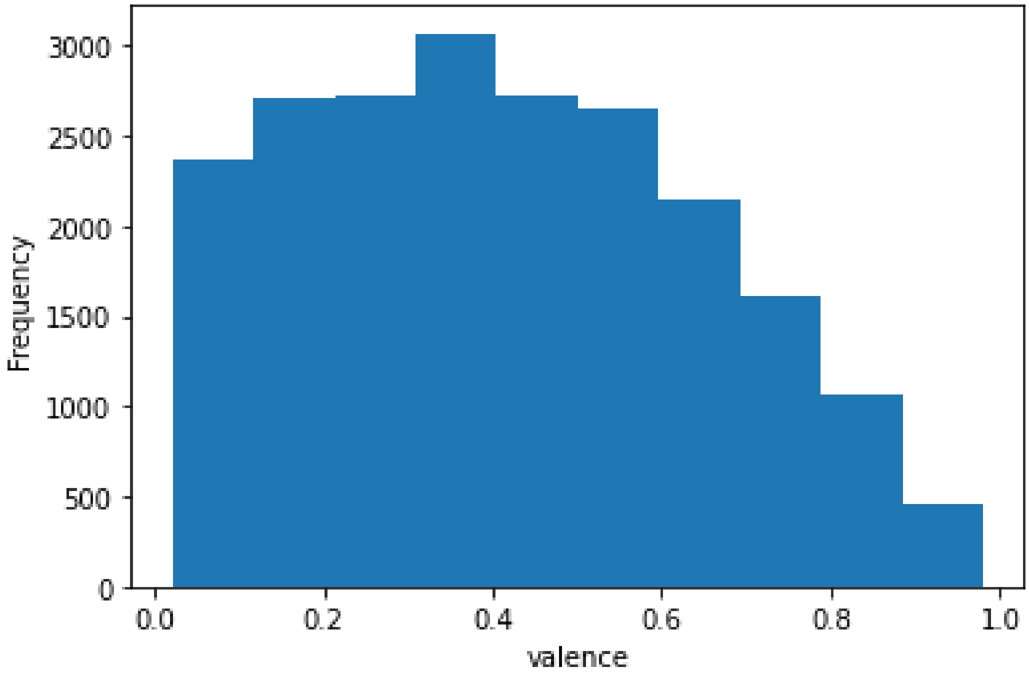
Songs also have a high energy to them in the dataset.



As mentioned before songs don’t use ‘instrumentalness’ as much

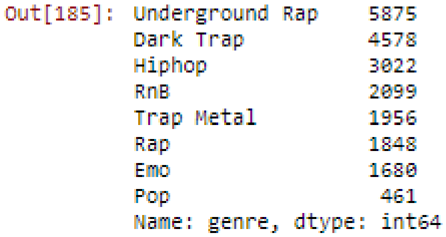


‘liveness’ is used more than ‘instrumentalness’ but still not as much.



Songs have a nice spread of ‘valence’

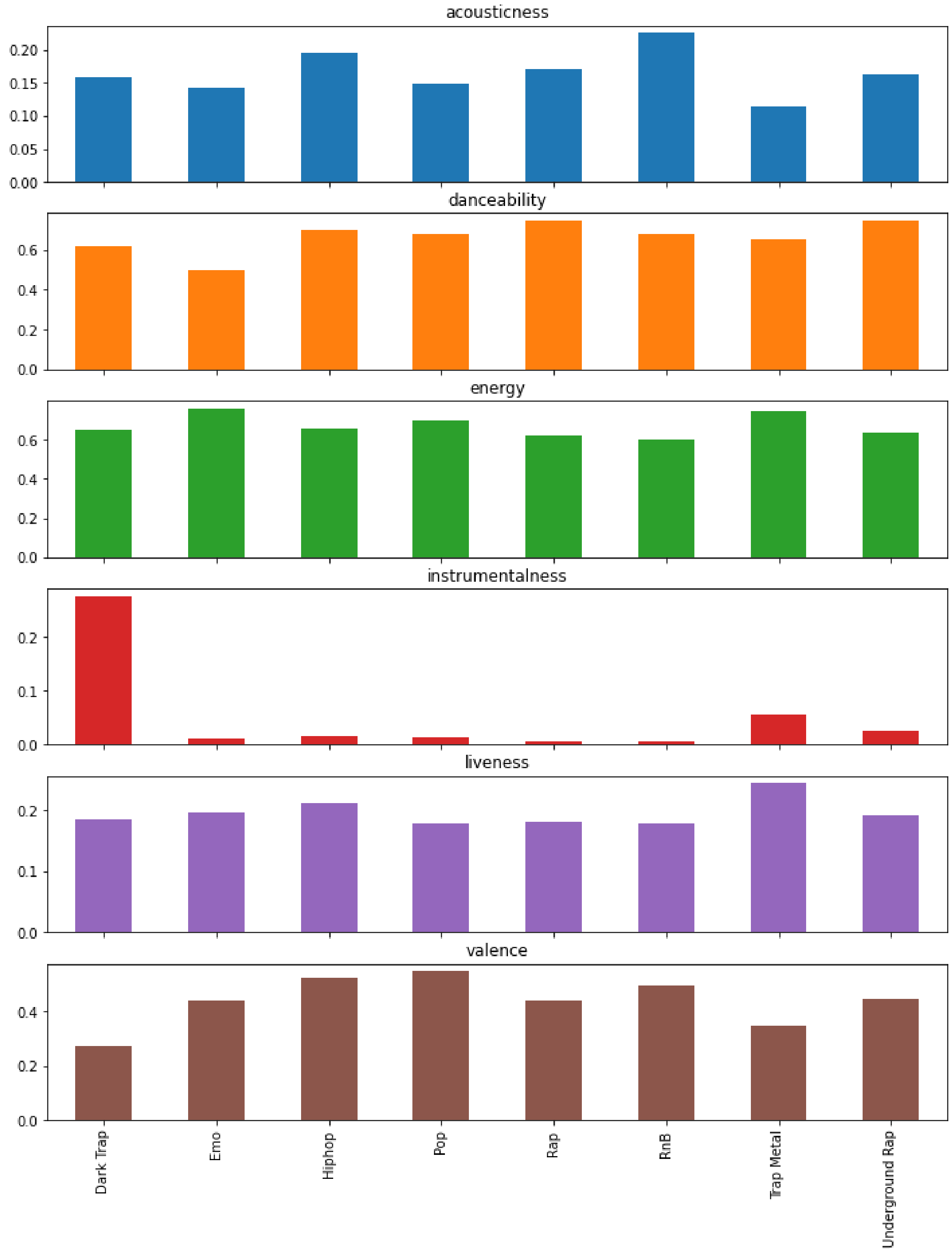
After I looked at the genre and took a value count to see what genre are being shown in the dataset.



```
Out[185]: Underground Rap    5875
          Dark Trap          4578
          Hiphop             3022
          RnB                2099
          Trap Metal         1956
          Rap                1848
          Emo                1680
          Pop                 461
          Name: genre, dtype: int64
```

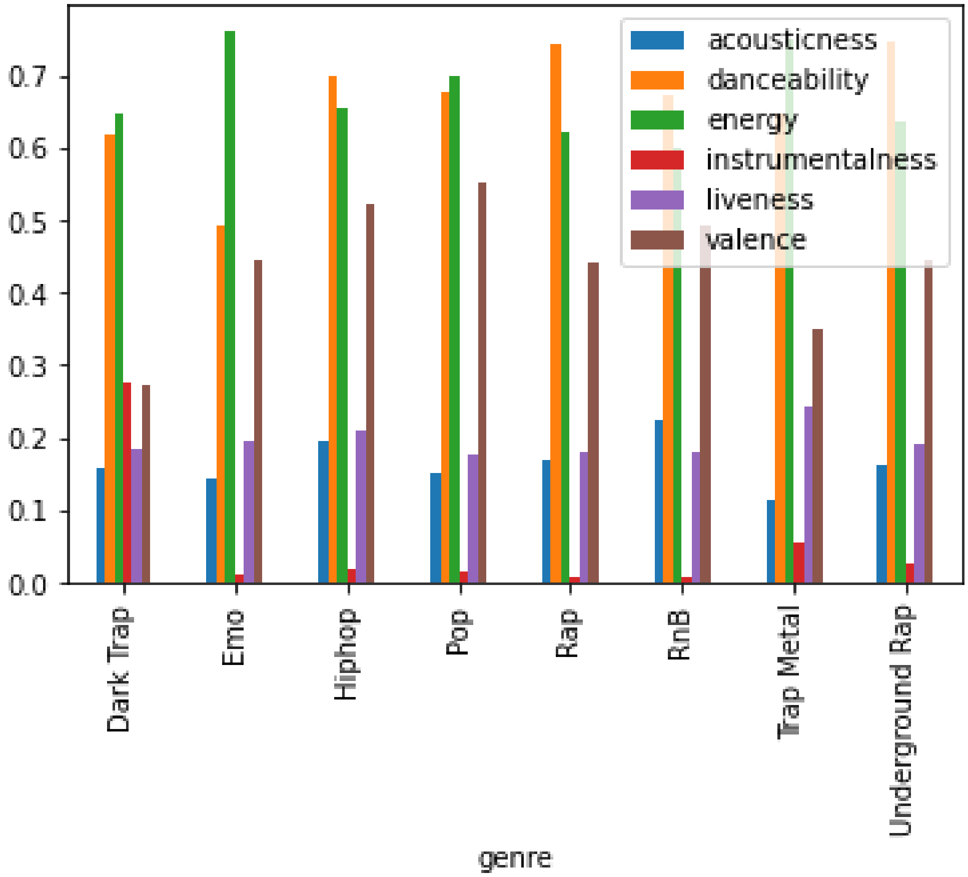
There weren’t many genres provide in the dataset, but there was a lot of songs in the genre that were provided. After grouping songs by its genre, I wanted to see if different genres had difference important on the sound features tested above.

Average of sound feature by genre



One thing that stood out was that Dark Trap genre used ‘instrumentalness’ much more than the rest of the genres and that, naturally, RnB was the highest for ‘acousticness’.

Another condensed graph showing the previous one:



Pre-Processing and Modeling:

To create the recommendation system, I wanted to convert the genres into genre clusters using Standard Scaler and then Kmeans, with 10 cluster values. After I used TNSE to dimension reduce the genre and clusters and then re-cluster with Standard Scalar and Kmeans.

I then had functions to find the given songs with genre, genre was needed to differentiated cause multiple songs had the same name. A function to get the mean vector of the given songs based on its sound features, then the recommendation function to suggest songs.

Future Research:

After doing this project I realized wished I picked a better dataset where the artist was included in it, it would of made for easier search and recommendation results. I also had to pruned some data that was double up, for example a song name ‘Rock your Body’ was given twice for two different genres, although it was the same song, it ruined the mean vector algorithm.