# Ghazal Rafiei
# Clustering Spotify Songs

**Student no. 97222044**

**Date: Dec 10, 2021**

**Subject: Data Science**

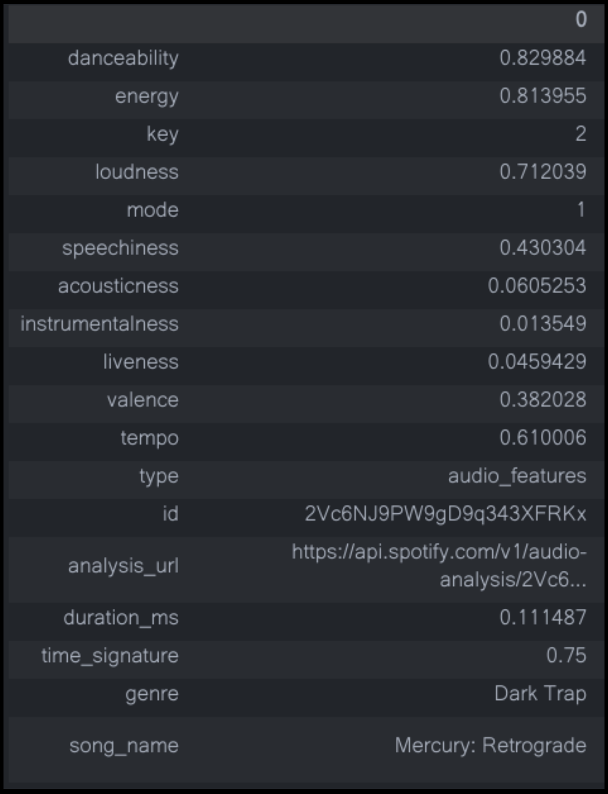
**Supervisor: Dr. Saeedreza Kheradpisheh**

## 1. Insight

The data is a collection of Spotify songs records. It consists of 42305 rows and 22 columns. Below, you can see the names of the columns.

*Danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, type, id, uri, track_href, analysis_url, duration_ms, time_signature, genre* and *song_name*.

Fortunately, the data has no null value, so there is no need to retrieve them. Table 1 shows one row sample of the data:

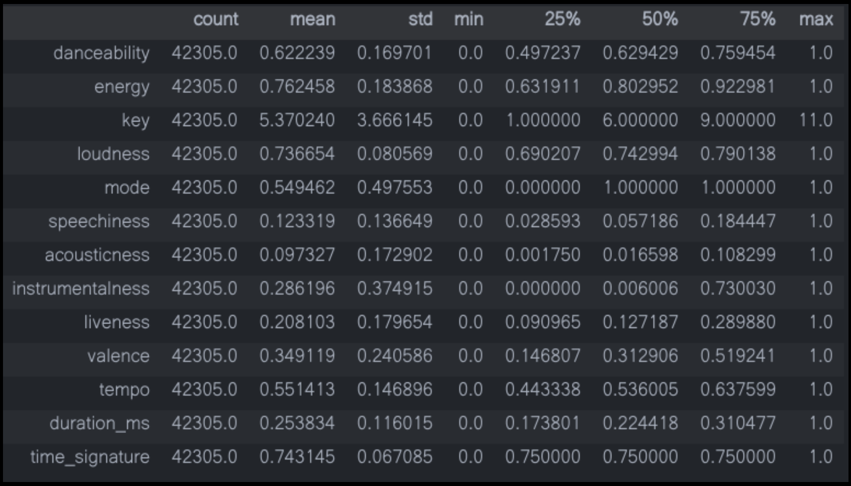


| | 0 |
|---|---|
| danceability | 0.829884 |
| energy | 0.813955 |
| key | 2 |
| loudness | 0.712039 |
| mode | 1 |
| speechiness | 0.430304 |
| acousticness | 0.0605253 |
| instrumentalness | 0.013549 |
| liveness | 0.0459429 |
| valence | 0.382028 |
| tempo | 0.610006 |
| type | audio_features |
| id | 2Vc6NJ9PW9gD9q343XFRKx |
| analysis_url | https://api.spotify.com/v1/audio-analysis/2Vc6... |
| duration_ms | 0.111487 |
| time_signature | 0.75 |
| genre | Dark Trap |
| song_name | Mercury: Retrograde |

**Table 1.** One row sample of data.

In this report, I am not going to exploit rows *analysis_url* and *track_h*f, so, I drop them.

Moreover, this a summary of our data:



| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| danceability | 42305.0 | 0.622239 | 0.169701 | 0.0 | 0.497237 | 0.629429 | 0.759454 | 1.0 |
| energy | 42305.0 | 0.762458 | 0.183868 | 0.0 | 0.631911 | 0.802952 | 0.922981 | 1.0 |
| key | 42305.0 | 5.370240 | 3.666145 | 0.0 | 1.000000 | 6.000000 | 9.000000 | 11.0 |
| loudness | 42305.0 | 0.736654 | 0.080569 | 0.0 | 0.690207 | 0.742994 | 0.790138 | 1.0 |
| mode | 42305.0 | 0.549462 | 0.497553 | 0.0 | 0.000000 | 1.000000 | 1.000000 | 1.0 |
| speechiness | 42305.0 | 0.123319 | 0.136649 | 0.0 | 0.028593 | 0.057186 | 0.184447 | 1.0 |
| acousticness | 42305.0 | 0.097327 | 0.172902 | 0.0 | 0.001750 | 0.016598 | 0.108299 | 1.0 |
| instrumentalness | 42305.0 | 0.286196 | 0.374915 | 0.0 | 0.000000 | 0.006006 | 0.730030 | 1.0 |
| liveness | 42305.0 | 0.208103 | 0.179654 | 0.0 | 0.090965 | 0.127187 | 0.289880 | 1.0 |
| valence | 42305.0 | 0.349119 | 0.240586 | 0.0 | 0.146807 | 0.312906 | 0.519241 | 1.0 |
| tempo | 42305.0 | 0.551413 | 0.146896 | 0.0 | 0.443338 | 0.536005 | 0.637599 | 1.0 |
| duration_ms | 42305.0 | 0.253834 | 0.116015 | 0.0 | 0.173801 | 0.224418 | 0.310477 | 1.0 |
| time_signature | 42305.0 | 0.743145 | 0.067085 | 0.0 | 0.750000 | 0.750000 | 0.750000 | 1.0 |

**Table 2.** Primary statics of the data.

I can infer from table 2 that most of the columns are normalized by default since their values are between 0.0 and 1.0. Although the *key* column has discrete variables, I normalized that to have all the data consistent.

## 2. Visualization

Now, I am going to gain some inference from the data via visualization.

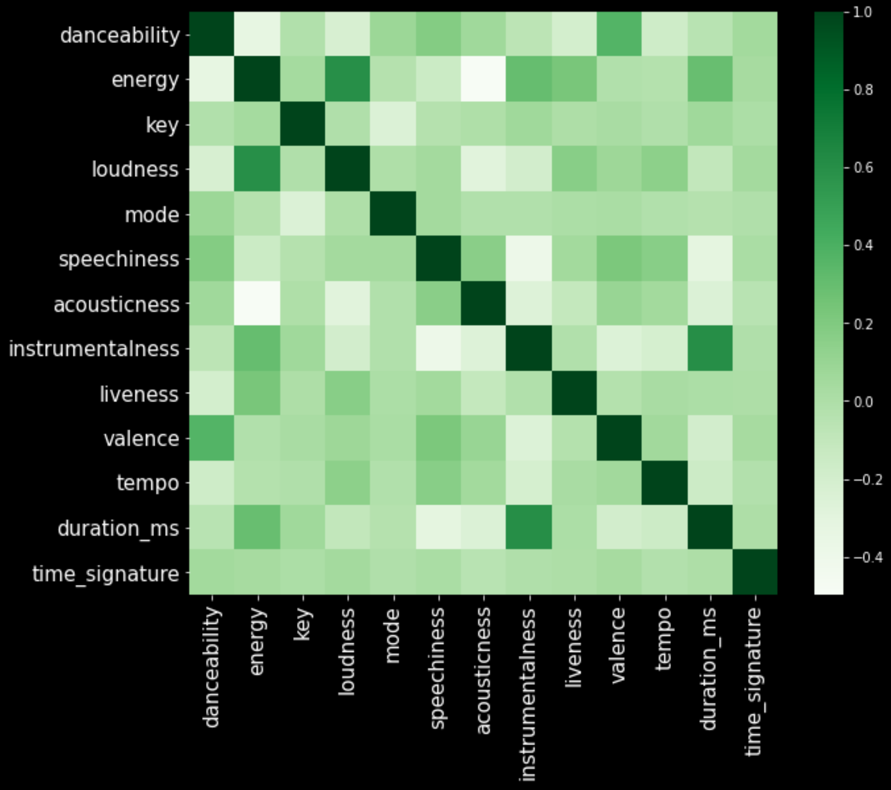


**Figure 1**. Correlation of every two columns.

As you can see in the figure 1, columns such as *instrumentalness*, *duration*, *loudness* and *energy* have a high correlation. On the other hand, *speechiness*, *instrumentalness*, *energy* and *acousticness* have a negative correlation which means they are inversely proportional.

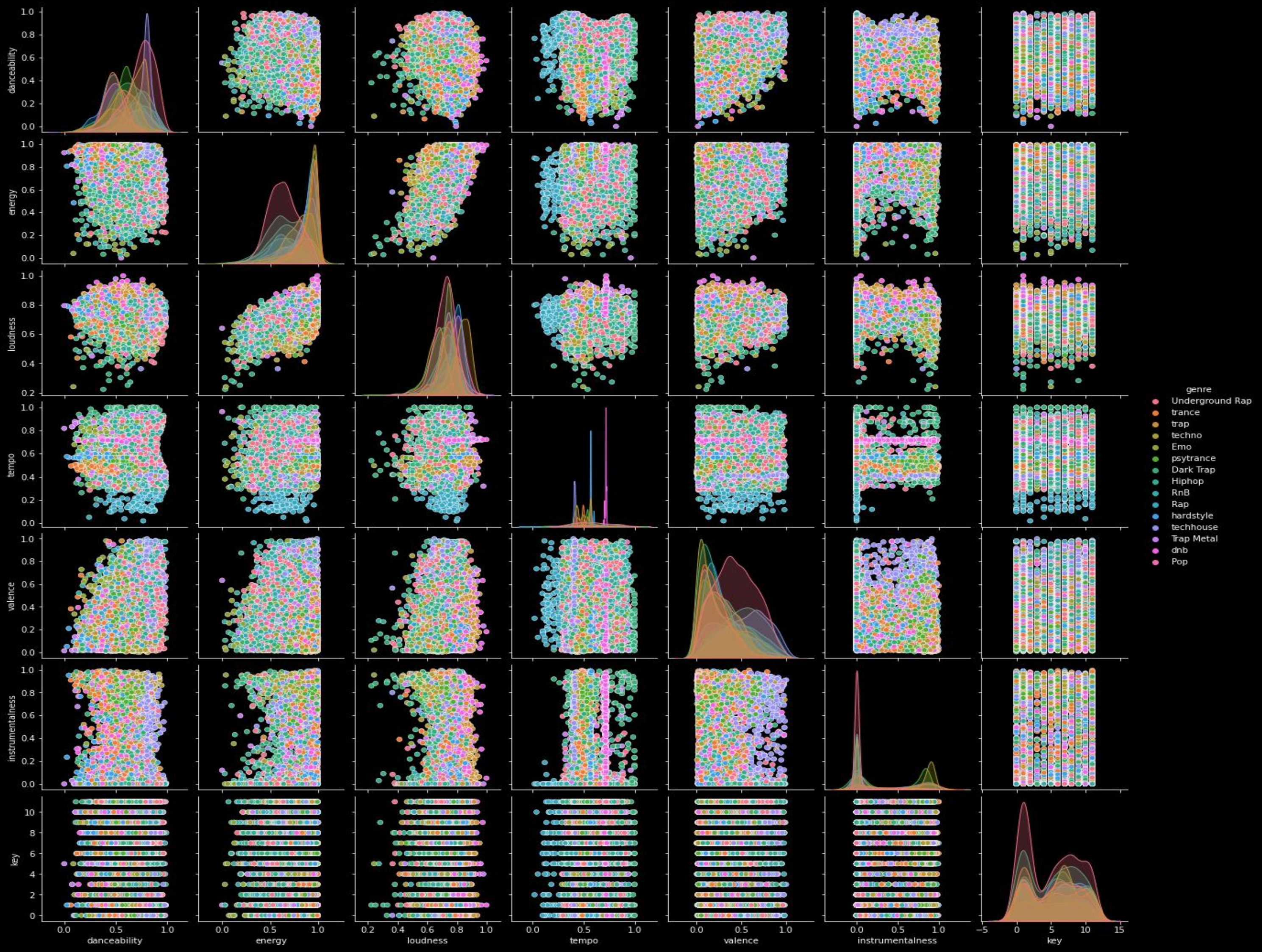


**Figure 2.** Pairplot of *tempo, valence, energy* and *danceability.*

In figure 2 you can see the correlation of some of the columns that can be related to each other. Also, in diagonal you can see the distribution of those features. For example, I can understand that *energy* and *loudness* are so related to each other while *tempo* and *valence* can be two independent variables.

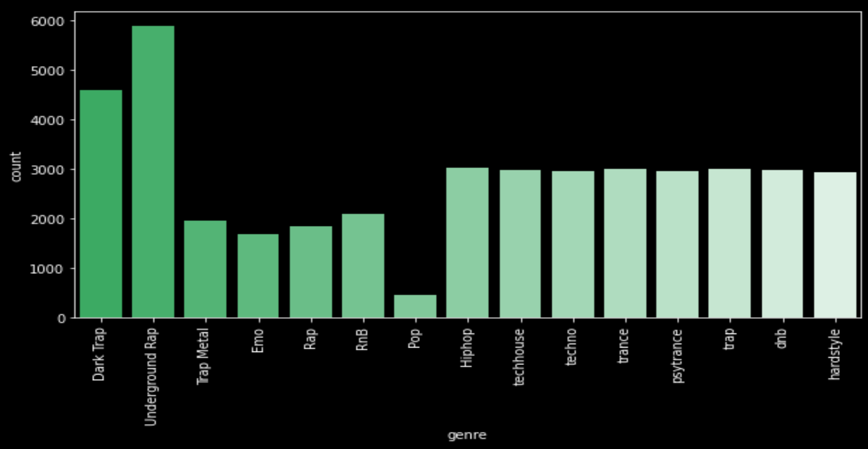


**Figure 3.** Histogram of genres distribution.

Figure 3 illustrates the distribution of different genres. Pop and underground pop songs respectively constitute the least and the most music genres in this dataset. All others are the same negligibly.

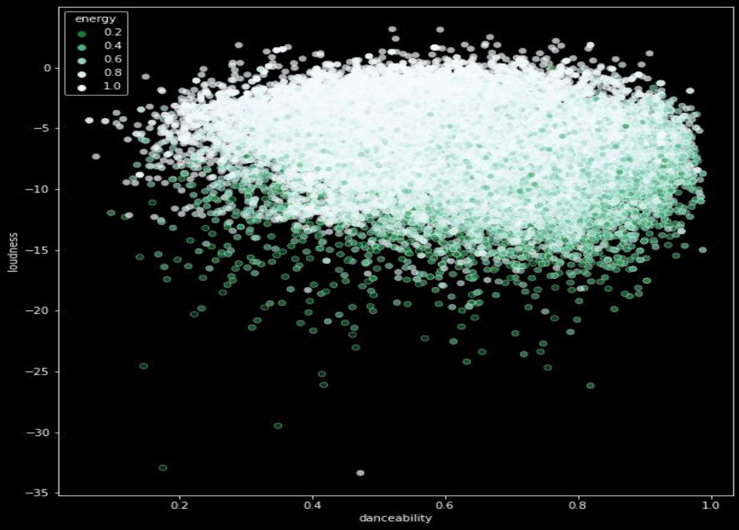


**Figure 4.** Relativity of *danceability* and *loudness* with *energy*

Figure 4 depicts that the louder the song has been played and the more it is dance-able, its energy is higher.

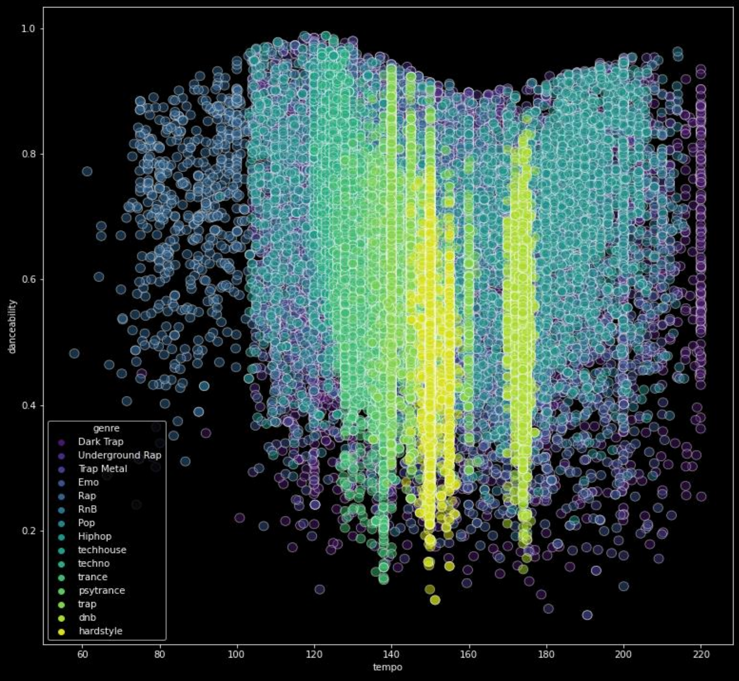


**Figure 5.** Relativity of *danceability* and *loudness* with *energy*

In figure 5, from the different colors of areas in the plot, I can realize that different genres have different danceability and these two columns should be effective in clustering.

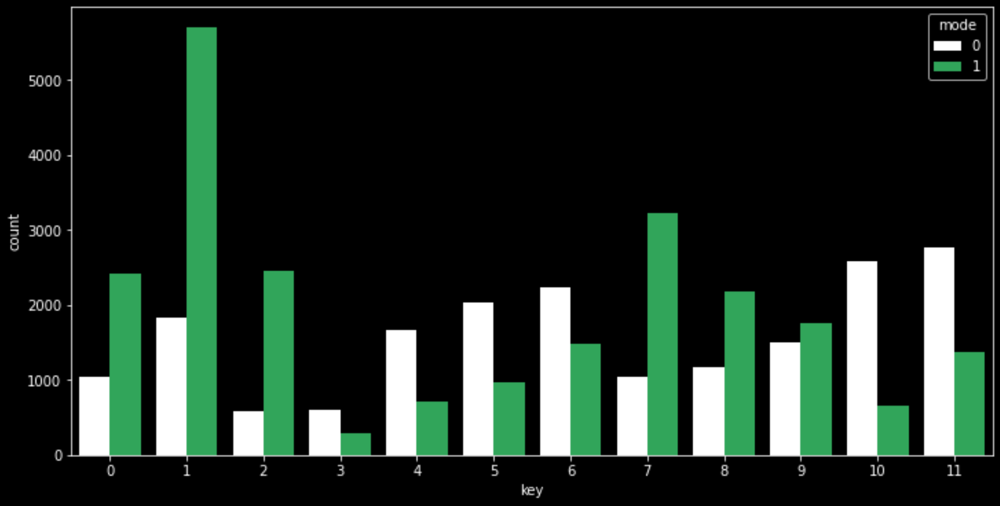


**Figure 6.** Count of minor and major keys in each key.

Also, in figure 6 I can see how many music are in major or minor key in each genre.

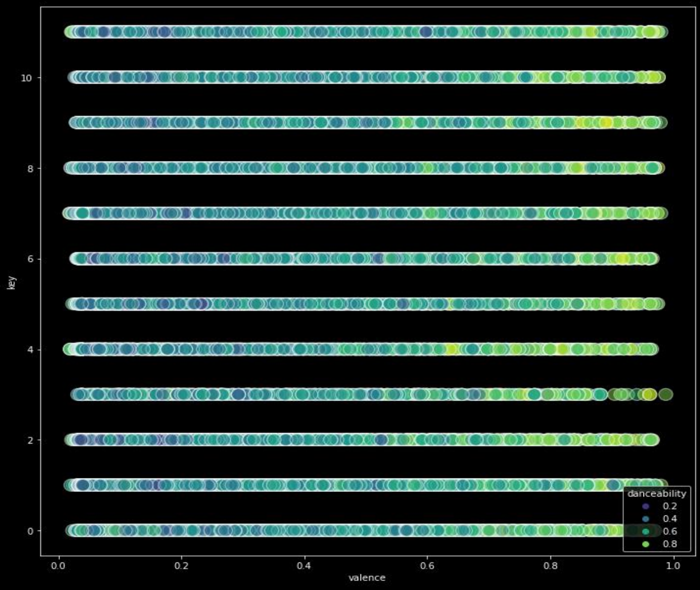


**Figure 6.** Relativity of *danceability* and *loudness* with *energy.*

Finally, figure 6 depicts that the songs with more valence, are more danceable as what I expected. However, there is no striking dependence between danceability and valence.

## 3. Preprocessing

For this stage, firstly I make sure all numeric columns are normalized for the reason that I want all columns to have the same impact in calculating distance and clustering. Regarding negative values, I do this by using the following formula:

$$(x - min(x))/(max(x) - min(x))\,. \tag{1}$$

Then, I choose these columns out of others for the reason that others may not contribute any value to data for clustering.
*danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo* and *genre.*

Also, I exploit feature extraction to reduce the dimension of data. In this stage, 11 columns are diminished to 5.

## 4. Processing

I detach 10 percent of the data for the future testing and only work with the remaining 90 percent.

For clustering, I tested Kmeans, DBSCAN, HDBSCAN and Spectral clustering. Unfortunately, only Kmeans was runnable on my local computer and others crashed because the program would fill the whole 16GB memory! Therefore, I continued with Kmeans and different count of clusters. I tested 5 to 20 number of clusters. Unsurprisingly, the more clusters we have, the less the error becomes, because the data points will be closer to the centre of their cluster is case of more clusters.

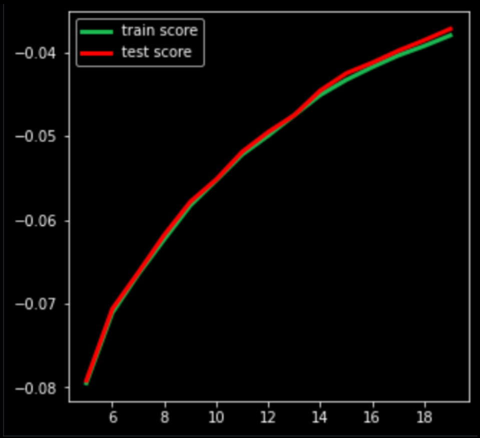


**Figure 7.** The change of train and test score on clusters come from *Kmeans.*

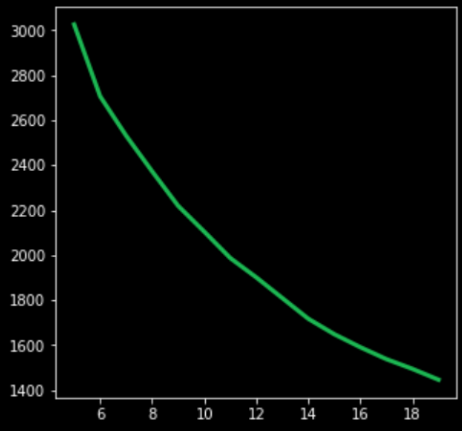


**Figure 8.** Inertia[1] of train set over different number of clusters.

Figure 8 is indicative of this statement. Thus, I ought to use another metric to find the best *k*. I opted out to find the elbow of this graph using Kneed [2]package in python. Finally, *k* was chosen 11.

Below, in figure 9, you can see the distribution of genres among clusters. Despite our expectation, clusters are not pre-empted to each genres and

[1] Sum of squared distances to the centre of belonging cluster of each data point.

[2] https://github.com/arvkevi/kneed

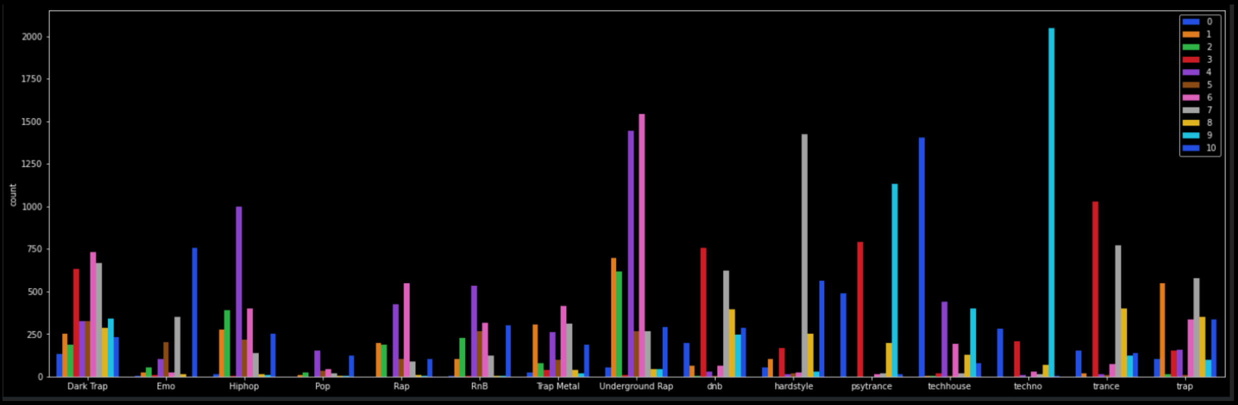


**Figure 9.** Distribution of clusters among genres.

genre, same as other columns, is only one factor in choosing clusters. However, we can see for example the cluster 9 mostly includes the genre Techno or underground rap songs constitute the large proportion of the clusters 5 and 6.

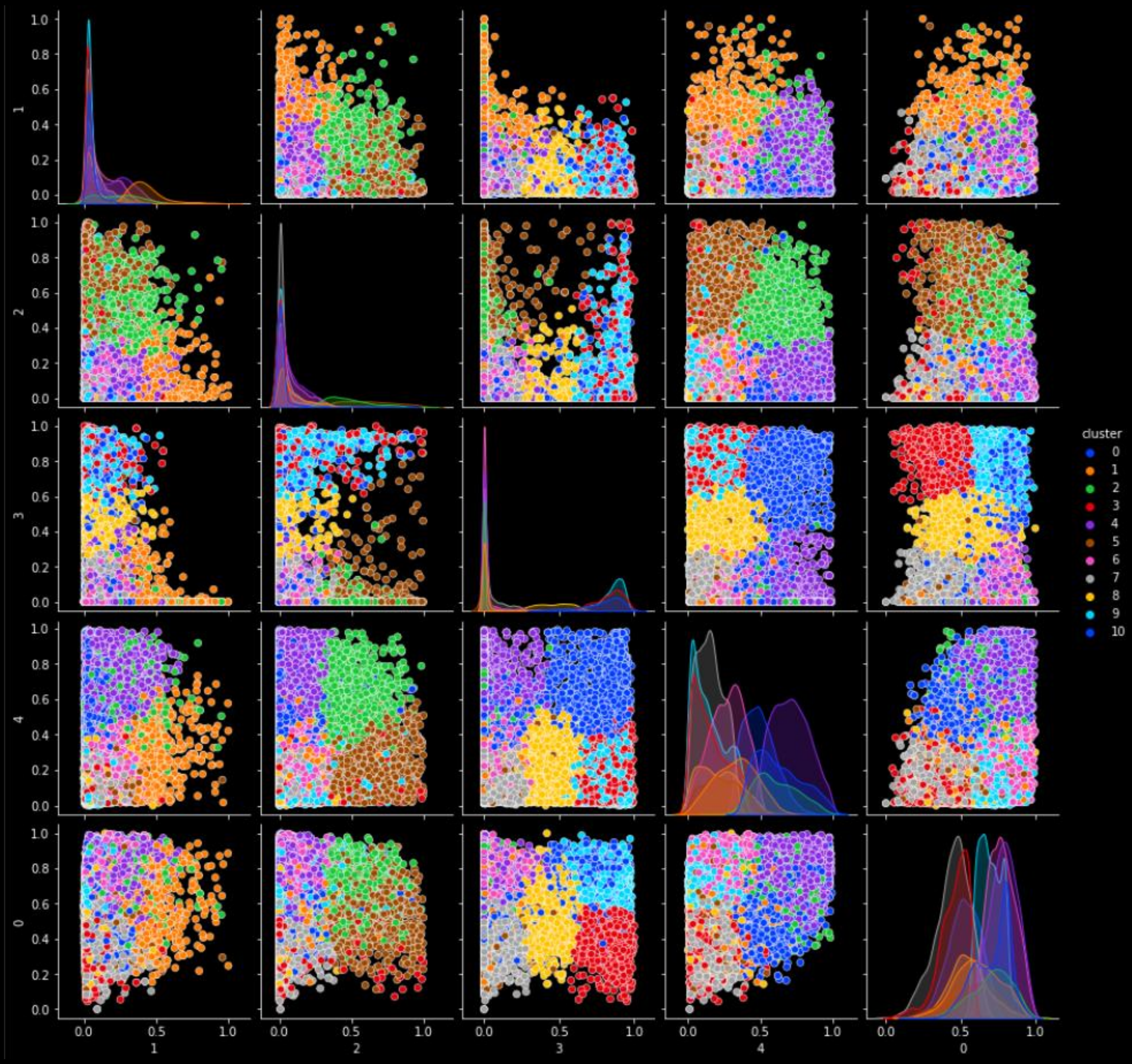


**Figure 10.** Pairplot of data hued by number of cluster.

To put an end to the result evaluation, I drew this pair plot in figure 10 which contains 5 features on both x and y axis which are extracted from feature selection and they are hued by number of clusters. In almost all of this 20 non-diagonal scatter plots and we can see that data points in each cluster are somehow separated. The reason why we cannot see all of these 11 colors in each plot, is the real dimension of data is 5 and when we picture them in 2 dimensions, some groups overlap and only few of them are observable in each view.

## 5. Testing

100 random row was taken out from the data frame and saved in test.csv.

For the testing stage, all of the data transformations such as normalization and dimension reduction were carried out on the test. Then the above model used to predict their cluster.

## 6. Recommendation

Firstly, the number of clusters the users mostly listen was sort and the 5 topmost extracted. Then 5 random songs from each one of them was saved into *Daily Mix.csv* file. If the users did not listen 5 clusters, the remaining was chosen from random cluster to be suggested to the user.

Secondly, the two closest songs to each centre of clusters, was saved in *top songs.csv* file.

## 7. Resources

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html