Movies Dataset Report

Akshat Shukla

February 11, 2025

Abstract

An analysis on the Song dataset provided by KDAG, IIT Kharagpur as a part of their Selection Process

1 Generate vectors for the three keywords

1.1 Dataset overview

The dataset we are provided with has the following columns:

- Song ID
- Genre (GT Values)
- keyword1
- keyword2
- keyword3

Song ID	Genre (GT Values)	keyword1	keyword2	keyword3
74	guitar	happy	distorted	rock
103	brass	energetic	melodic	classical
201	banjo	happy	acoustic	country
194	synth	energetic	heavy	hip-hop
184	synth	energetic	slow	hip-hop
97	brass	calm	upbeat	classical
63	guitar	energetic	melodic	rock
...	...	...	...	...

Table 1: Head of our Dataframe

These keywords correspond to different aspects of the song. We will generate vectors for these keywords using the TF-IDF method as well as the Bag of words method.

for our use, the TF-IDF Embedding model captures more information since it takes into account the frequency of the words in the dataset. The Bag of Words model only captures the presence of the word in the dataset. Since both of these correspond to embeddings with the same dimensions, the TF-IDF Model is considered for our use.

1.2 Keyword Embedding Generation

To generate the keyword embeddings, we will use the TF-IDF method. The TF-IDF (Term Frequency-Inverse Document Frequency) method helps in understanding the importance of a keyword in relation to the dataset.

keyword_1	
guitar	65
synth	43
piano	12
brass	11
violin	10
banjo	6
Name: count, dtype: int64	

Table 2: Keyword 1 counts

keyword_2	
happy	30
mellow	28
energetic	27
sad	21
angry	12
emotional	11
calm	11
upbeat	4
nostalgic	3
Name: count, dtype: int64	

Table 3: Keyword 2 counts

keyword_3	
fast	28
melodic	27
slow	23
upbeat	20
rhythmic	14
heavy	10
acoustic	9
twangy	6
distorted	5
danceable	5
Name: count, dtype: int64	

Table 4: Keyword 3 counts

1.2.1 TF-IDF Calculation

The TF-IDF value is calculated as follows:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t) \tag{1}$$

where:

- $\text{TF}(t, d)$ is the term frequency of term t in document d.
- $\text{IDF}(t)$ is the inverse document frequency of term t.

The term frequency (TF) is calculated as:

$$\text{TF}(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{2}$$

where $f_{t,d}$ is the frequency of term t in document d.

The inverse document frequency (IDF) is calculated as:

$$\text{IDF}(t) = \log\left(\frac{N}{|\{d \in D : t \in d\}|}\right) \tag{3}$$

where N is the total number of documents and $|\{d \in D : t \in d\}|$ is the number of documents containing term t.

We generate the TF-IDF Bases Embeddings for each keyword in the dataset and store them for future use

[4]

Song ID	tf-idf-keyword1	tf-idf-keyword2	tf-idf-keyword3
74	[0., 0., 0.36, 0., 0., 0.]	[0., 0., 0., 0., 0.32, 0., ...]	[0., 0., 0.11, 0., 0., 0., ...]
103	[0., 0.19, 0., 0., 0., 0.]	[0., 0., 0., 0.31, 0., 0., ...]	[0., 0., 0., 0., 0., 0.31, ...]
201	[0.13, 0., 0., 0., 0., 0.]	[0., 0., 0., 0., 0.32, 0., ...]	[0.17, 0., 0., 0., 0., 0., ...]
194	[0., 0., 0., 0., 0.35, 0.]	[0., 0., 0., 0.31, 0., 0., ...]	[0., 0., 0., 0., 0.18, 0., ...]
...	...	...	...

Table 5: Generated Embeddings

2 Dimensionality Reduction

We will utilize Principal Component Analysis (PCA) on individual keywords in our dataset in order to reduce the dimensionality of the generated embeddings.

Via Principal Component analysis, we find the directions along which the variance of the data is maximized and project the data onto these directions.

Thus, we can reduce the dimensionality of the data while retaining most of the information.

The Principle Component vectors are found as:

$$\text{PCA}(X) = X \cdot V \tag{4}$$

where X is the standardized data matrix and V is the matrix of eigenvectors of the covariance matrix of X. [2]

We will use the first 2 eigenvectors to reduce the dimensionality of the data.

song_id	kw1_pca1	kw1_pca2	kw2_pca1	kw2_pca2	kw3_pca1	kw3_pca2
74	1.349694	0.370477	-2.051477	0.344695	-0.047019	-0.192033
103	-0.538837	-1.813324	0.695438	-1.910713	-1.646868	1.482767
201	-0.501082	-1.376430	-2.051477	0.344695	-0.055551	-0.233782
194	-1.556300	1.033730	0.695438	-1.910713	-0.058292	-0.247738
184	-1.556300	1.033730	0.695438	-1.910713	-0.203708	-1.817350

Table 6: PCA values for keywords

We use `np.cov` in order to obtain the covariance matrix of the given standardized data and use `np.linalg.eigh` in order to obtain the eigenvectors of the covariance matrix

3 Combining the Embeddings into one

3.1 Embedding Format

Here, we choose a simple embedding for every datapoint: The embedding is a three dimensional vector consisting of the average PCA values of the three keywords for that datapoint.

This provides us with a three dimensional view of each datapoint based on the values of their individual keywords.

The Embeddings were plottted using desmos here

4 Clustering

Here, we utilize k means clustering in order to cluster the embeddings into different groups. Later, we find out the silhouette score for various values of embeddings and plot them.

We choose a value of `k = 8` since the points, plotted in 3d space could be deciphered visually into 8 clusters.

Later, We will use these clusters in order to predict the Genre (Ground Truth) values for new datapoints.

To Implement k-means clustering, we initialize `k = 8` points in the space and assign each point to the cluster with the nearest centroid. We then update the centroid of each cluster to be the mean of the points in that cluster. We repeat this process until the centroids converge.

After clustering, all points are assigned clusters. Here, we can infer that the points belonging to the same cluster have the same genre, with the mode of each cluster appearing more than half of the times.

[1]

A more interactive simulation can be found over at desmos here

5 Analysis

5.1 Ground Truth Distribution in clusters

One comeback of this approach is that we were unable to assign any clusters with the `country` genre as the modal one since the number of such datapoints do not appear to be close together in the 3d space.

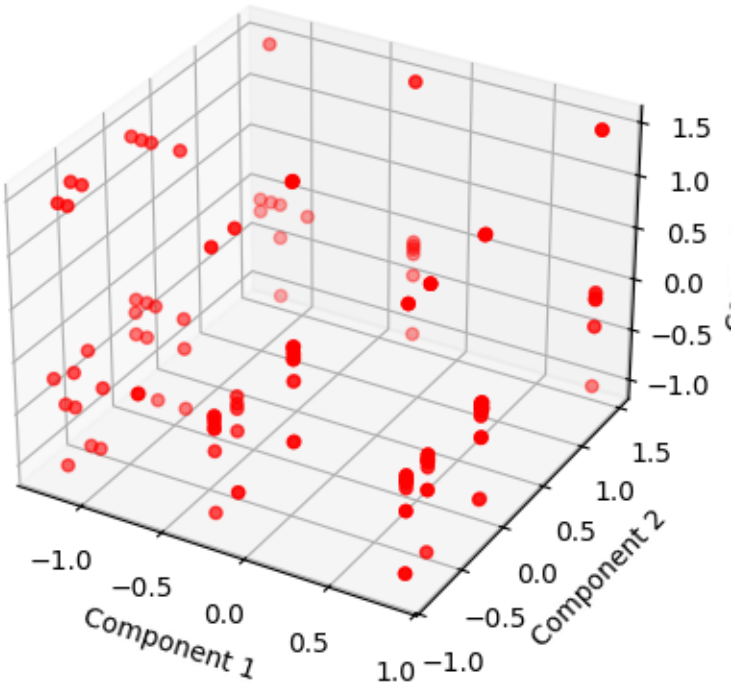


Figure 1: Visualization of the combined embeddings

Cluster Number	Modal Genre	# Points	Percentage of Modal Genre
0	rock	45	44.44
1	rock	22	22.73
2	hip-hop	15	66.67
3	rock	8	50.00
4	hip-hop	15	66.67
5	classical	14	50.00
6	classical	22	68.18
7	pop	6	33.33

Table 7: Cluster Analysis

Even though there are overlaps, we were able to seperate almost all the genres into different clusterings.

5.2 Silhouette Score

The silhouette score is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from -1 to 1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.

The silhouette score for a given clustering is given as:

$$\text{Silhouette Score} = \frac{b - a}{\max(a, b)} \tag{5}$$

where:

- a is the mean distance between a sample and all other points in the same class.
- b is the mean distance between a sample and all other points in the next nearest cluster.

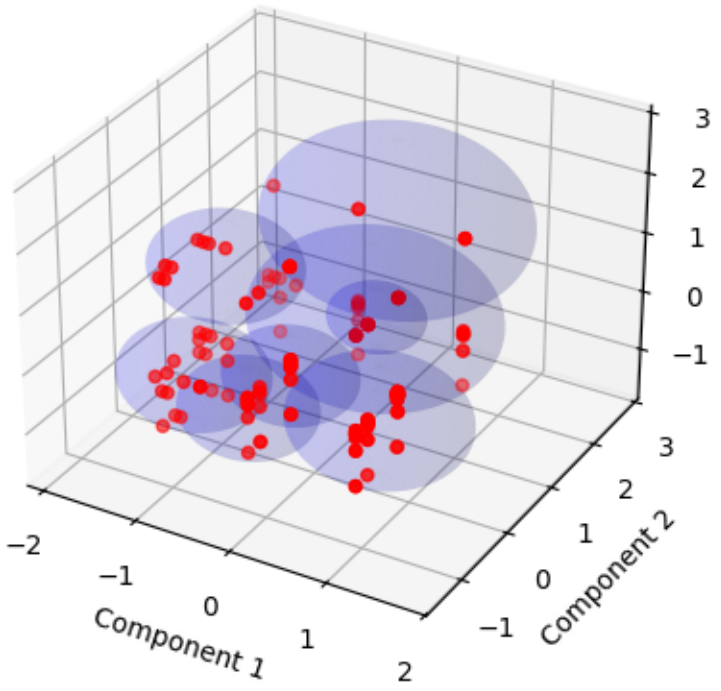


Figure 2: Visualization for the final cluster assignment

[3]

The total silhouette score is equal to the mean of all the silhouette scores of the individual clusters.

For given `k = 8`, We were able to obtain a Silhouette score of `0.79`

We plotted the Silhouette score for various values of k and found that the score kept increasing as the number of clusters increased. This is because the number of clusters increased, the points were more likely to be closer to their own cluster than to the next nearest cluster.

5.3 Assigning Genres to new datapoints

- In order to assign genres to new data, we perform exactly the same operations onto them as we did when generating embeddings for our original dataset, using the means and standard deviations of our original data set.
- We find out the closest cluster for this new data and assign the genre for the datapoint to be picked based on a binomial distribution on the genres present in the cluster

Point Number	Keyword 1	Keyword 2	Keyword 3	Closest Cluster	Assigned Genre
0	piano	calm	slow	6	classical
1	guitar	emotional	distorted	0	rock
2	synth	mellow	distorted	1	rock

Table 8: Assigning clusters to new datapoints

The assigned genres are: classical, country and rock respectively. which seem to be intuitive based on the keywords provided.

An interactive simulation can be viewed on desmos here

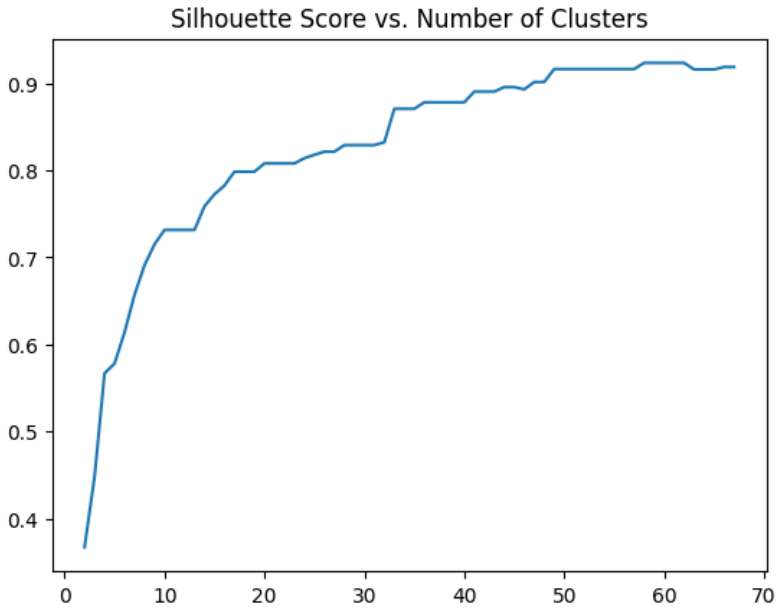


Figure 3: Silhouette Score for various values of k

6 Bonus

6.1 Extrinsic measures for clustering

some of the extrinsic measures used for analyzing the validity of clustering are:

- **The Fowlkes-Mallows index** It is defined as:

$$\text{FMI} = \frac{\text{TP}}{\sqrt{(\text{TP}+\text{FP}) \times (\text{TP}+\text{FN})}} \tag{6}$$

  where:

  - TP is the number of true positives (pairs of points that are in the same cluster and in the same ground truth class).
  - FP is the number of false positives (pairs of points that are in the same cluster but in different ground truth classes).
  - FN is the number of false negatives (pairs of points that are in the same ground truth class but in different clusters).

  [5]

- **Rand Index** used to measure the similarity between the ground truth and the clustering. The Rand Index is defined as:

$$\text{RI} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}} \tag{7}$$

  where:

  - TP is the number of true positives (pairs of points that are in the same cluster and in the same ground truth class).
  - FP is the number of false positives (pairs of points that are in the same cluster but in different ground truth classes).
  - FN is the number of false negatives (pairs of points that are in the same ground truth class but in different clusters).

– TN is the number of true negatives (pairs of points that are in different clusters and in different ground truth classes).

[6]

Here, we utilize the Rand Index in order to find the similarity between the ground truth and the clustering. The Rand Index ranges from 0 to 1, where a value of 1 indicates that the clustering is identical to the ground truth.

The Rand Index is calculated using the labels predicted using a binomial distribution and ground truth labels. We obtain a Rand Index of `0.734` which indicates that the clustering replicates the ground truth to a large extent.

song_id	genre	assigned_genre
74	rock	rock
103	classical	classical
201	country	pop
194	hip-hop	hip-hop
184	hip-hop	pop

Table 9: Comparison of Actual and Assigned Genres

6.2 Ingenious Embeddings for the keywords

- We utilize embedding by rank. The embedding for each keyword is a vector of length `6` where the value at each index corresponds to the rank of the keyword in the count of the keyword in the dataset.
- We find the optimal value of k by finding out the shilouette score at various values. I turns out that we have a local maxima at `k = 6` and thus we choose this value for our clustering.

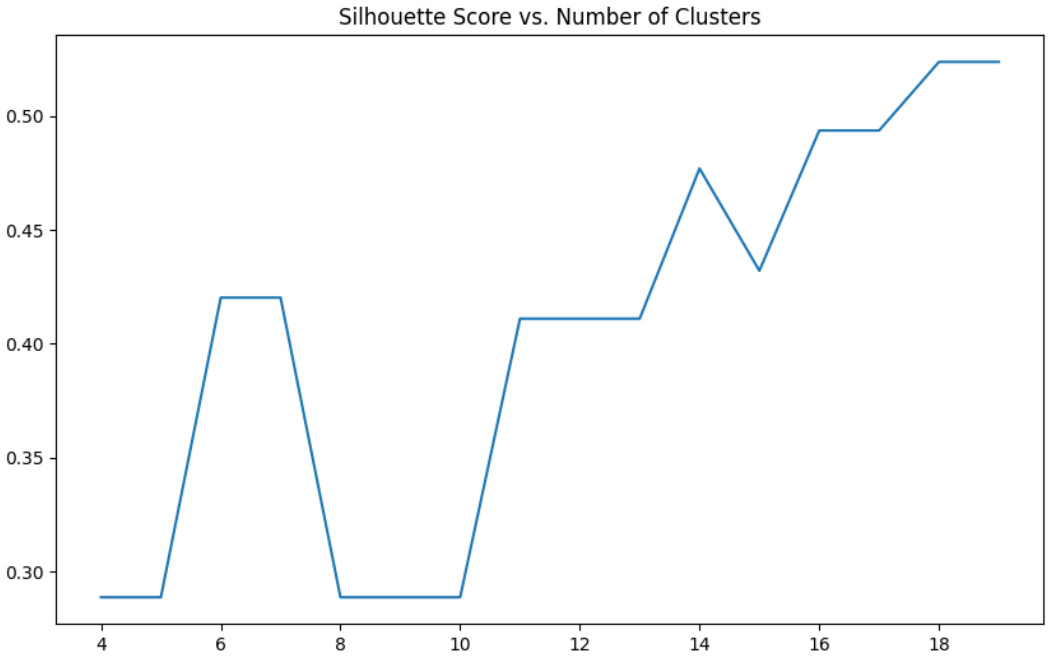


Figure 4: Silhouette score for the clustering across various k

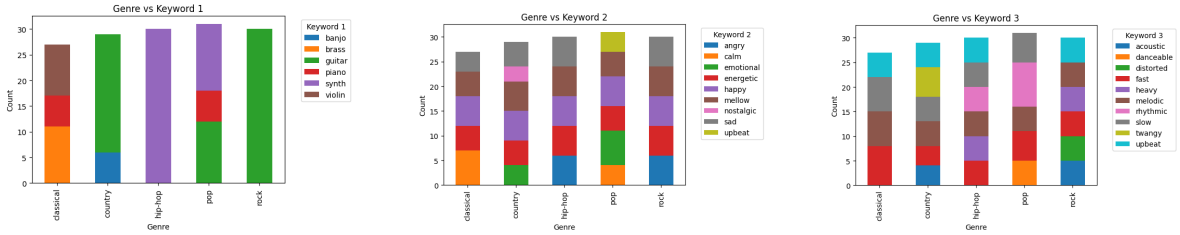
- However, for this clustering, the Rand Index turns out to be `0.66` which is much worse than what we had achieved earlier. This clearly depicts the requirement of a good embedding for the clustering to be successful.

- Visually, it is tough to separate these embeddings into groups. The embeddings generated can be visualized here
- Clearly, Clustering based on the embeddings generated by the TF-IDF method is much better than the embeddings generated by the rank method.

6.3 Further exploration of the Dataset

6.3.1 Analyzing the relation between genres and keywords

- Looking at the bar plot for Genre v/s Keyword 1, It becomes clear that the instrument being used largely affects the genre of the song.
- For example, all hip-hop make use of `synth`. All rock songs make use of `guitar`. Looking closely at the data, only classical songs make use of the `brass` or `violin`.
- Looking at Genre v/s Keyword 2, the separation is not very clear. However, some inferences can be drawn. For example, if a song is `upbeat`, it is more likely to be a `pop` song. If a song is `calm`, it is more likely to be a `classical` or a `pop` song.



(a) Genre vs Keyword 1 (b) Genre vs Keyword 2 (c) Genre vs Keyword 3

Figure 5: Analysis of Genre vs Keywords

6.3.2 Analyzing the Correlation between Keywords and genres

- The Correlation matrix for the keywords and genres brings out similar inferences as were drawn from the bar plots.
- For example, `guitar` and `rock` have a high correlation of `0.57`. Moreover, `hip-hop` and `synth` have The highest correlation of `0.79`.

See Figure 6.

6.3.3 Analyzing the Keyword-Keyword Correlation

- The correlation matrix for the keywords brings out some interesting inferences.
- For example, `calm` and `guitar` are strongly related.
- pairs (`mellow`, `melodic`), (`fast`, `mellow`), (`fast`, `energetic`) are strongly related
- pairs (`melodic`, `guitar`), (`fast` are also strongly related
- these inferences align with the perception for these keywords in the real world.

See Figure 7.

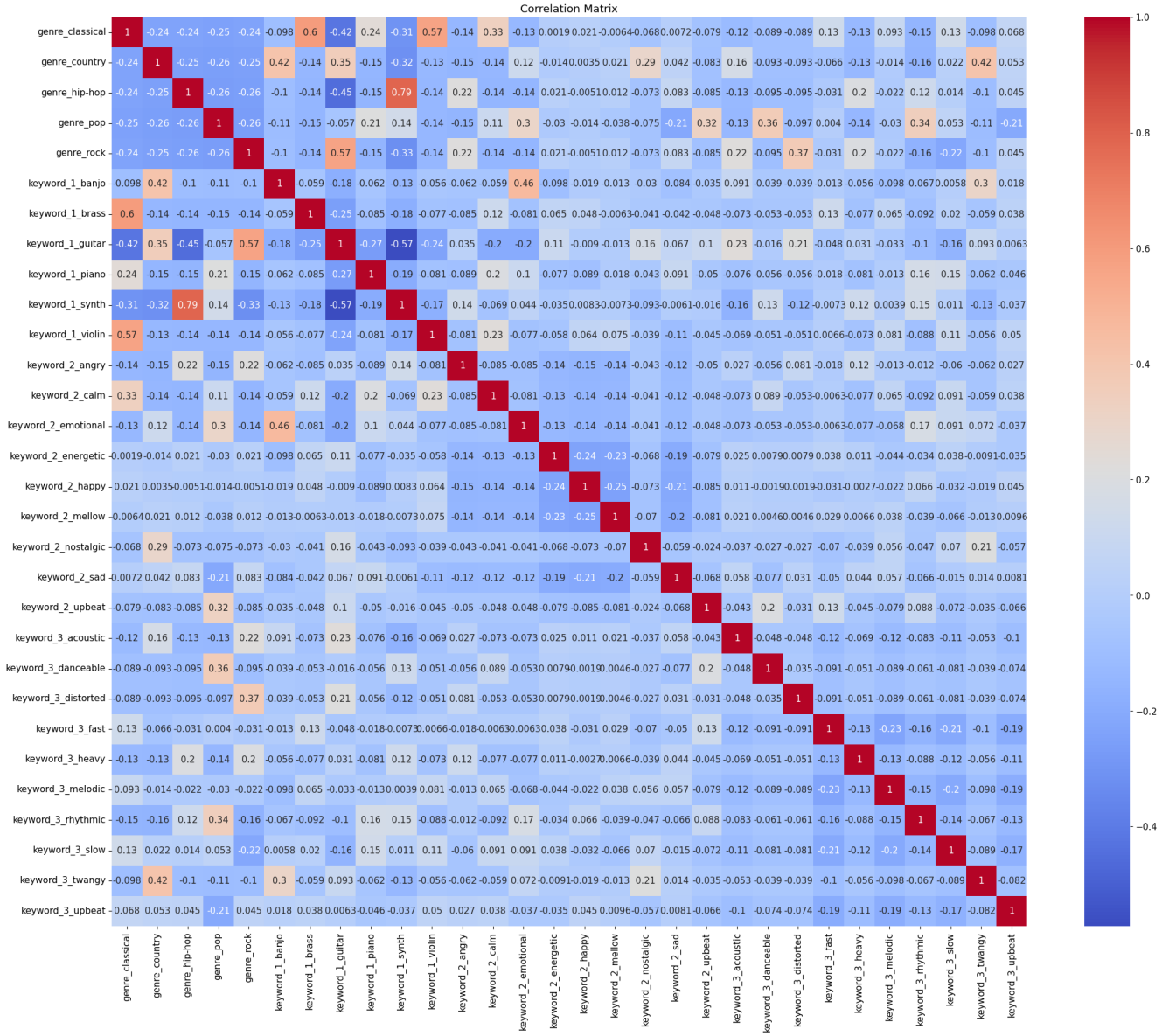


Figure 6: Correlation Matrix for Keywords

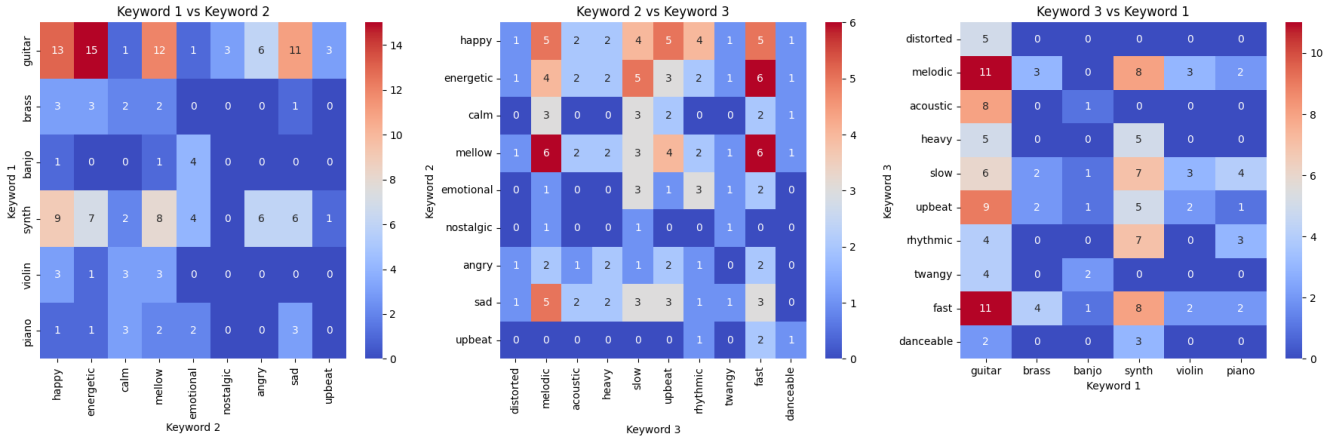


Figure 7: Correlation Matrix for Keywords

References

[1] GeeksforGeeks. K-means clustering introduction, 2020. Accessed: 2023-10-10.

[2] GeeksforGeeks. Principal component analysis (pca), 2020. Accessed: 2023-10-10.

[3] GeeksforGeeks. Silhouette algorithm to determine the optimal value of k, 2020. Accessed: 2023-10-10.

[4] Towards Data Science. Word embedding techniques: Word2vec and tf-idf explained, 2020. Accessed: 2023-10-10.

[5] Wikipedia. Fowlkes-mallows index, 2020. Accessed: 2023-10-10.

[6] Wikipedia. Rand index, 2020. Accessed: 2023-10-10.