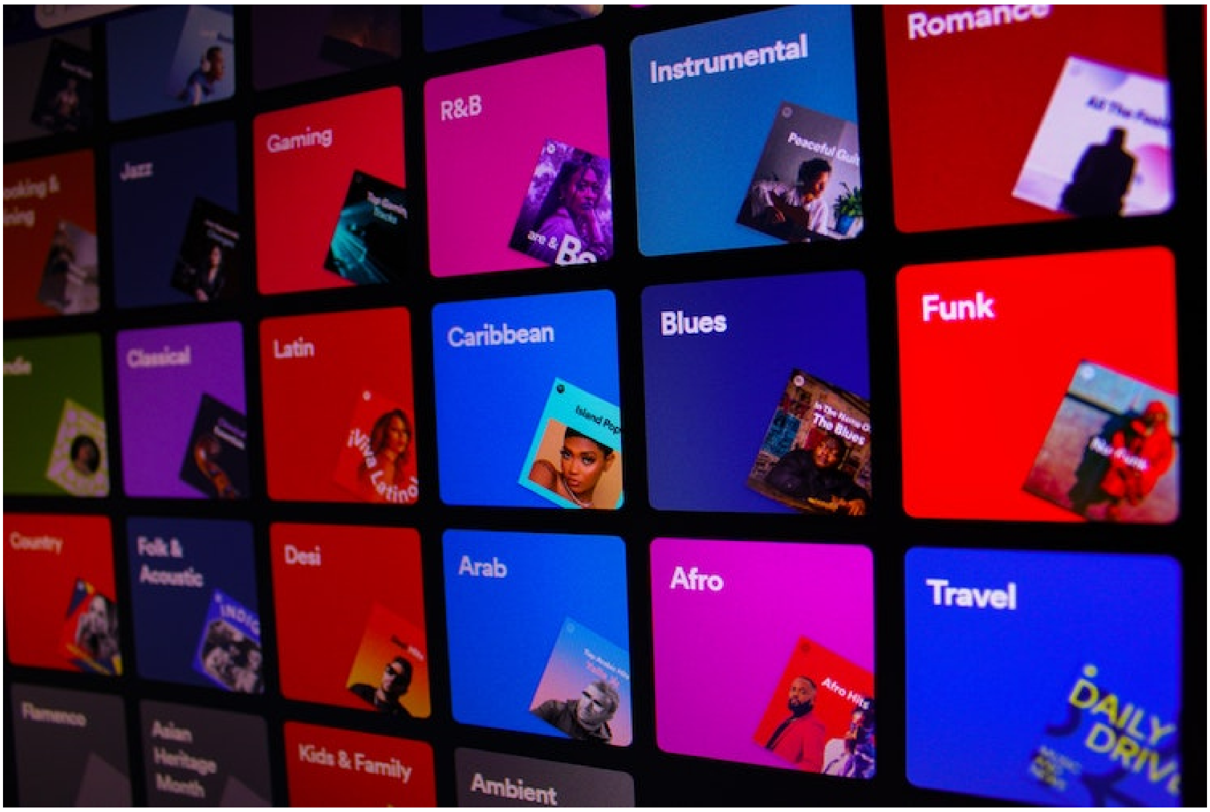
Music Recommendation System Using SVD, Feature Projection, and Dimensionality Reduction

Feolu Kolawole

March 7, 2025



Abstract

Music is something everyone can feel. Defining such feelings to a universal standard has proved difficult, and as a result, we have classified music by genre. It goes without saying that the expectation is for a genre to yield similar "feelings" of music, and to a large extent that is true. However, what if we were to expand our horizon and look at music from a more analytical point of view? What if our classification became based on features like valence, energy, and tempo? Would such a classification result in similar genre-based definitions, or would the classification of music be revamped? This project uses singular value decomposition (SVD) to discover latent features in a data set of songs, create a weighted recommendation system based on reduced-dimensionality musical features, and explore emerging classifications of music.

1 Dataset

The dataset consists of songs extracted from Spotify with the following audio features:

- Danceability
- Energy
- Speechiness
- Acousticness
- Instrumentalness
- Liveness
- Valence
- Loudness
- Tempo

These features are normalized using z-score normalization to ensure that all features contribute equally to the similarity calculations:

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

where x is the original feature value, μ is the mean of the feature, and σ is the standard deviation.

id	artist_names	track_name	source	key	mode	time_signature	danceability	energy	speechiness	acousticness	instrumentalness	liveness	valence	loudness	tempo
000xQL6tZNLJzIrtIgxqSl	ZAYN, PARTYNEXTDOOR	Still Got Time (feat. PARTYNEXTDOOR)	RCA Records Label	G	Major	4 beats	0.748	0.627	0.0639	0.131	0.0	0.0852	0.524	-6.029	120.963
003eoIwxETJujVWmNFMoZy	Alessia Cara	Growing Pains	Def Jam Recordings	C#/Db	Minor	4 beats	0.353	0.755	0.733	0.0822	0.0	0.39	0.437	-6.276	191.153
003vvx7Niy0yvhvHt4a68B	The Killers	Mr. Brightside	Island Records	C#/Db	Major	4 beats	0.352	0.911	0.0747	0.00121	0.0	0.0995	0.236	-5.23	148.033
00B7TZ0Xawar6NZ00JFomN	Cardi B, Chance the Rapper	Best Life (feat. Chance The Rapper)	Atlantic/KSR	A	Major	4 beats	0.62	0.625	0.553	0.287	0.0	0.314	0.665	-7.438	167.911
00BIm7zeNqgYLPtW6zg8cj	Post Malone, The Weeknd	One Right Now (with The Weeknd)	Republic Records	C#/Db	Major	4 beats	0.687	0.781	0.053	0.0361	0.0	0.0755	0.688	-4.806	97.014
00EPIEnX1JFjff8sC6bccd	Thalia, NATTI NATASHA	No Me Acuerdo	Sony Music Latin	G	Minor	4 beats	0.836	0.799	0.0873	0.187	0.0	0.092	0.772	-4.247	94.033
00ETaeHUQ6lops3oWU1Wrt	Kygo, Donna Summer	Hot Stuff	RCA Records Label	F	Major	4 beats	0.681	0.773	0.148	0.019	1.28E-06	0.11	0.429	-5.749	119.961
00ZKeP47bZtswtANkvxz2j	Tropa do Bruxo, DJ Ws da Igrejinha, SMU, Triz, Mc Menor Thalis	Baile do Bruxo	Tropa Do Bruxo	G	Minor	5 beats	0.734	0.228	0.53	0.889	0.00642	0.102	0.522	-4.731	162.524
00gpGR84M27moP7AFuqHlx	YBN Nahmir	Bounce Out With That	2018	G#/Ab	Major	4 beats	0.857	0.56	0.173	0.0426	0.0	0.153	0.482	-8.278	94.949
00imgaPlYRrMGn9o83hfmk	Brent Faiyaz	LOOSE CHANGE	Lost Kids LLC., Marketed by Venice / Stem	C#/Db	Minor	4 beats	0.574	0.369	0.0814	0.753	0.0	0.147	0.44	-8.931	84.975
00ko9WaS4jOX1kEk3gvHjf	Chance the Rapper, MadeinTYO, DaBaby	Hot Shower	Chance the Rapper	A	Major	4 beats	0.899	0.509	0.387	0.00157	0.0	0.0573	0.599	-8.04	150.001

Figure 1: Example tracks from the Spotify dataset with audio features

2 SVD for Feature Projection

2.1 Mathematical SVD Background

Singular Value Decomposition (SVD) is a powerful matrix factorization technique that decomposes a matrix into three component matrices. For any matrix A of dimensions $m \times n$, SVD expresses it as:

$$A = U\Sigma V^T$$

Where:

- U is an $m \times m$ orthogonal matrix containing the left singular vectors
- Σ is an $m \times n$ diagonal matrix containing the singular values in descending order
- V^T is the transpose of an $n \times n$ orthogonal matrix containing the right singular vectors

To understand how these matrices are computed, we start with the matrix $A^T A$, which is always symmetric. The spectral theorem guarantees that any symmetric matrix has an orthonormal basis of eigenvectors, which allows us to write:

$$A^T A v_i = \sigma_i^2 v_i$$

where v_i are the eigenvectors and σ_i^2 are the eigenvalues of $A^T A$. These v_i vectors form the columns of matrix V.

For the left singular vectors, we observe that when we multiply A by v_i:

$$A v_i = \sigma_i u_i$$

where $u_i = \frac{A v_i}{\sigma_i}$ if $\sigma_i \neq 0$. These u_i vectors are orthonormal and form the columns of matrix U. If the rank of A is r, then only the first r singular values σ_i are positive, while the rest are zero. This gives us the complete decomposition:

$$A = U \Sigma V^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

The singular values are arranged in descending order: $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$.
This decomposition provides us with orthonormal bases for fundamental subspaces of A:

- The columns of V corresponding to non-zero singular values form an orthonormal basis for the row space of A
- The columns of U corresponding to non-zero singular values form an orthonormal basis for the column space of A

These orthonormal bases are what make SVD ideal for dimensionality reduction and projection.

2.2 Why Feature Projection Matters

When we analyze music using audio features like danceability, energy, and tempo, we're working with high-dimensional data that may contain noise. Features that may not be so significant can influence our song recommendations. To combat this we can use feature projection through SVD, which allows us to discover latent features—underlying patterns that aren't explicitly represented in the original data.

These projected features can capture musical characteristics more meaningful than individual audio metrics. For example, rather than considering danceability and energy separately, a projected feature might represent a combination that corresponds to "club-worthiness" or "workout intensity." In mathematical terms this corresponds to a linear combination of features for projection. In our case, by using SVD, we essentially create a linear combination that acts as a weighting system where the coefficients are based on the strength of the singular values. For example:

$$\text{Projected Feature}_1 = 0.7 \times \text{danceability} + 0.6 \times \text{energy} - 0.3 \times \text{acousticness} + 0.2 \times \text{tempo}$$

The orthonormal basis provided by SVD ensures that each latent feature captures a unique, uncorrelated aspect of musical variation. This creates a more efficient representation of musical similarity that better aligns with human perception.

2.3 Projection Implementation

To implement feature projection, I first compute the SVD of our audio features matrix. I then select the top k components that capture the most variance in the data, typically choosing a value much smaller than the original feature dimension works best (in our case, $k = 5$). The projection is calculated as:

$B_k = B \cdot V_k$

Where:

- B is the original audio features matrix
- V_k is the matrix containing the first k columns of V
- B_k is the projected representation of songs in the reduced k-dimensional space

This projection operation maps songs from their original feature space onto the orthonormal basis defined by the top k right singular vectors. It may seem strange that we claim projection but seem to perform matrix-vector multiplication, but the following shows that in our case they are equivalent.

When projecting a vector $\mathbf{v}$ onto an orthonormal basis $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k\}$, the standard projection formula is:

$$\text{proj}_{\{\mathbf{w}_1,\ldots,\mathbf{w}_k\}}(\mathbf{v}) = (\mathbf{v} \cdot \mathbf{w}_1)\mathbf{w}_1 + (\mathbf{v} \cdot \mathbf{w}_2)\mathbf{w}_2 + \cdots + (\mathbf{v} \cdot \mathbf{w}_k)\mathbf{w}_k$$

Or in summation form:

$$\sum_{i=1}^{k}(\mathbf{v} \cdot \mathbf{w}_i)\mathbf{w}_i$$

If we treat all vectors $\mathbf{w}_i$ as columns of a matrix W, we can rewrite the dot products as a matrix multiplication. When $\mathbf{v}$ is treated as a row vector and multiplied by W:

$$\mathbf{v} \times W = [(\mathbf{v} \cdot \mathbf{w}_1), (\mathbf{v} \cdot \mathbf{w}_2), \ldots, (\mathbf{v} \cdot \mathbf{w}_k)]$$

This gives us the coordinates of $\mathbf{v}$ in the basis $\{\mathbf{w}_i\}$. For our feature projection, these coordinates represent our songs in the reduced latent feature space.
Thus, for our entire dataset B, the projection simplifies to:

$$B_k = B \cdot V_k$$

The k value is chosen based on the singular values in Σ, which indicate how much variance each component explains. We select enough components to capture a substantial portion of the total variance while still achieving significant dimensionality reduction.

Finally, I use cosine similarity to rank the projections and take the top similarities. (Further Elaboration in 3.3).

2.4 Geometric Visualization

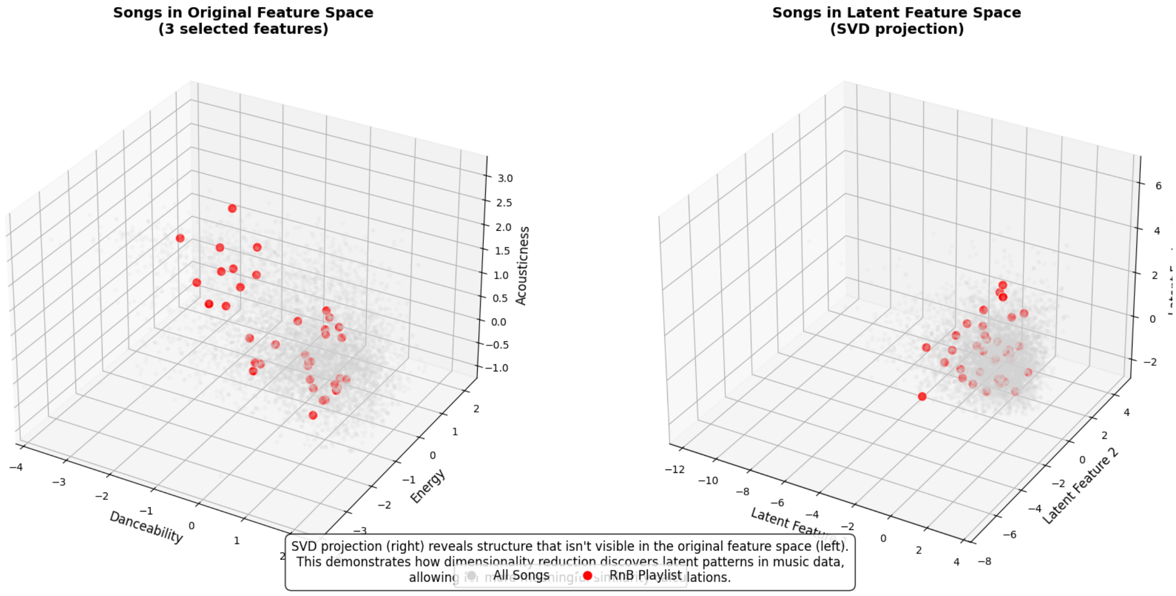


Figure 2: Projection of songs into latent feature space using SVD

The figure demonstrates the benefits of SVD-based dimensionality reduction in the context of our music recommendation system. Here, we can see songs projected from their original 9-dimensional space (danceability, energy, acousticness, etc.) into a 3-dimensional space discovered through SVD.

The red dots represent songs from the R&B playlist, while the gray dots show the entire song database. Notice how the R&B songs form distinct clusters within this reduced space, revealing patterns that weren't clear in the original high-dimensional representation.

Each axis in this visualization represents a feature - a linear combination of the original audio features. The proximity of songs in this reduced space directly informs our recommendation algorithm. Calculating the cosine similarity between song vectors in this latent space compares them based on these discovered musical patterns rather than raw features, resulting in recommendations that better capture the feel of the music.

3 Recommendation Approaches

3.1 One-Per-Song Method

The One-Per-Song method is a recommendation strategy that prioritizes diversity by finding one similar song for each track in the input playlist. This approach ensures that each song in the original playlist contributes to the final recommendations, regardless of how similar or dissimilar it might be to other songs. The algorithm works as follows:

1. For each song in the input playlist, compute its similarity with all songs in the database
2. Sort these similarities in descending order
3. Select the most similar song that hasn't already been added to the recommendations
4. Add this song to the recommendation list
5. Continue until we have one recommendation for each input song

3.2 Overall Top Method

The Overall Top method takes a more global approach to recommendation by finding songs with the highest similarity scores across the entire input playlist. Instead of ensuring each input song contributes, this method identifies the most similar songs overall.
The algorithm proceeds as follows:

1. Calculate similarities between every input song and every database song
2. Collect all these similarity scores into a single list
3. Sort the list by similarity score in descending order
4. Select the top songs, avoiding duplicates
5. Return the resulting recommendation list

3.3 Comparison of Methods

While both comparison methods make use of the same SVD and Feature-Projection algorithm, they do offer tradeoffs.

- **Diversity vs. Similarity**: One-Per-Song ensures diversity at the potential cost of including some less similar songs, while Overall Top maximizes similarity at the risk of lacking variety.
- **Playlist Coherence**: Overall Top tends to produce more coherent recommendations that strongly match the dominant elements of the input playlist.

The optimal approach depends on the user's intent. One-Per-Song works better for diverse input playlists where users want to explore variations of each song. Overall Top is better for focused playlists where users want more of exactly what dominates their current selection.

4 Example

4.1 Example Input Playlist

To demonstrate our recommendation system, we can utilize a initial playlist of songs. This initial selection serves as the base for our algorithm to identify and recommend music with similar characteristics and mood.

Input Playlist
Song: No Love (with SZA), Artist: Summer Walker, SZA
Song: Slime You Out (feat. SZA), Artist: Drake, SZA
Song: Just Us (feat. SZA), Artist: DJ Khaled, SZA
Song: Kill Bill (feat. Doja Cat), Artist: SZA, Doja Cat
Song: What Lovers Do (feat. SZA), Artist: Maroon 5, SZA
Song: The Weekend - Funk Wav Remix, Artist: SZA, Calvin Harris, Funk Wav
Song: What Lovers Do (feat. SZA), Artist: Maroon 5, SZA
Song: Love Galore (feat. Travis Scott), Artist: SZA, Travis Scott
Song: All The Stars (with SZA), Artist: Kendrick Lamar, SZA
Song: Kill Bill, Artist: SZA
Song: Saturn, Artist: SZA
Song: Love Language, Artist: SZA
Song: I Do (feat. SZA), Artist: Cardi B, SZA

Song: TELEKINESIS (feat. SZA & Future), Artist: Travis Scott, SZA, Future
Song: Rich Baby Daddy (feat. Sexyy Red & SZA), Artist: Drake, Sexyy Red, SZA
Song: Blind, Artist: SZA
Song: Low, Artist: SZA
Song: All The Stars (with SZA), Artist: Kendrick Lamar, SZA
Song: Gone Girl, Artist: SZA
Song: I Hate U, Artist: SZA
Song: F2F, Artist: SZA

4.2 SVD Analysis and Feature Interpretation

Before we get to the the recommendations, let's examine how our SVD analysis decomposes the audio features into patterns:

Top Singular Values:

- Singular value 1: 126.85
- Singular value 2: 89.32
- Singular value 3: 85.21
- Singular value 4: 80.53
- Singular value 5: 79.57

Latent Feature Linear Combinations:

Latent Feature 1:

$$0.279 \times \text{danceability} + 0.545 \times \text{energy} + 0.055 \times \text{speechiness} - 0.427 \times \text{acousticness} - 0.155 \times \text{instrumentalness} + 0.081 \times \text{liveness} + 0.354 \times \text{valence} + 0.526 \times \text{loudness} + 0.086 \times \text{tempo}$$

Latent Feature 2:

$$-0.602 \times \text{danceability} + 0.264 \times \text{energy} - 0.566 \times \text{speechiness} - 0.093 \times \text{acousticness} + 0.176 \times \text{instrumentalness} + 0.269 \times \text{liveness} - 0.282 \times \text{valence} + 0.234 \times \text{loudness} - 0.015 \times \text{tempo}$$

Latent Feature 3:

$$-0.228 \times \text{danceability} + 0.039 \times \text{energy} + 0.542 \times \text{speechiness} - 0.114 \times \text{acousticness} + 0.103 \times \text{instrumentalness} + 0.392 \times \text{liveness} - 0.190 \times \text{valence} - 0.079 \times \text{loudness} + 0.658 \times \text{tempo}$$

Latent Feature 4:

$$-0.006 \times \text{danceability} + 0.042 \times \text{energy} - 0.128 \times \text{speechiness} - 0.192 \times \text{acousticness} + 0.321 \times \text{instrumentalness} - 0.784 \times \text{liveness} - 0.149 \times \text{valence} + 0.059 \times \text{loudness} + 0.449 \times \text{tempo}$$

Latent Feature 5:

$$-0.253 \times \text{danceability} - 0.064 \times \text{energy} - 0.133 \times \text{speechiness} + 0.212 \times \text{acousticness} - 0.847 \times \text{instrumentalness} - 0.189 \times \text{liveness} + 0.028 \times \text{valence} + 0.095 \times \text{loudness} + 0.327 \times \text{tempo}$$

These equations represent what linear combinations our algorithm deemed best to provide recommendations based upon.

- **Latent Feature 1** can be interpreted as an "energy-loudness" dimension, with strong positive weights for energy (0.545) and loudness (0.526), contrasted with negative weights for acousticness (-0.427).
- **Latent Feature 2** shows strong negative weights for danceability (-0.602) and speechiness (-0.566), suggesting this dimension identifies songs that are less rhythmically dance-oriented and more melodically focused.
- **Latent Feature 3** emphasizes tempo (0.658) and speechiness (0.542), likely capturing the rhythmic and vocal delivery elements that are particularly important in genres like R&B and Rap.
- **Latent Feature 4** has heavy negative weighting for liveness (-0.784) and positive weighting for tempo (0.449).
- **Latent Feature 5** strongly downweights instrumentalness (-0.847), suggesting it identifies songs that are more vocal-focused.

4.3 Recommendation Results

When we process our input playlist through the algorithm using these new latent features, it generates the following collection of songs:

Song: Slime You Out (feat. SZA), Artist: Drake, SZA
Song: It's Nice To Have A Friend, Artist: Taylor Swift
Song: All The Stars (with SZA), Artist: Kendrick Lamar, SZA
Song: Cold (feat. Future), Artist: Maroon 5, Future
Song: Baby Pluto, Artist: Lil Uzi Vert
Song: Make Me (Cry), Artist: Noah Cyrus, Labrinth
Song: when the party's over, Artist: Billie Eilish
Song: MORE & MORE, Artist: TWICE
Song: What Lovers Do (feat. SZA), Artist: Maroon 5, SZA
Song: pete davidson, Artist: Ariana Grande
Song: Hold On, Artist: Adele
Song: You Are The Reason, Artist: Calum Scott
Song: Winter Wonderland, Artist: Tony Bennett
Song: City Of Stars - From "La La Land" Soundtrack, Artist: Ryan Gosling, Emma Stone
Song: I Hate U, Artist: SZA
Song: when the party's over - Recorded at Spotify Studios NYC, Artist: Lewis Capaldi
Song: What Lovers Do (feat. SZA), Artist: Maroon 5, SZA
Song: when the party's over, Artist: Billie Eilish
Song: Calling (Spider-Man: Across the Spider-Verse) (Metro Boomin & Swae Lee, NAV, feat. A Boogie Wit da Hoodie), Artist: Metro Boomin, Swae Lee, NAV, A Boogie Wit da Hoodie
Song: Primera Cita, Artist: Carin Leon
Song: RISE! (feat. DAISY WORLD), Artist: Tyler, The Creator, DAISY WORLD

4.4 Analysis of Results

The recommendations show several patterns revealed by the SVD-based approach:

- **Diverse genre representation:** Unlike the purely SZA-focused input playlist, our recommendations include artists from various genres that still yield mostly similar musical feeling (according to our latent features). Such analysis included pop (Taylor Swift, Ariana Grande), singer-songwriter (Billie Eilish, Adele), soundtrack music (Ryan Gosling, Emma Stone), and even Latin music (Carin Leon). It is important to note that this came from using an overall best-similarity approach as to prioritizing one most similar song per song in the input playlist. *Note: The second method yielded mainly SZA based results, which intuitively makes sense given a song sang by SZA is most likely to be more one-to-one similar to another song sang by SZA*

- **Selective artist continuity:** In contrast to the first bullet point, While the input playlist was dominated by SZA, the recommendations maintain some SZA tracks (*I Hate U, What Lovers Do, All The Stars*) but avoid overwhelming the results with a single artist, demonstrating the algorithm's ability to focus on sound qualities rather than simply matching artist names.

These results highlight how our SVD-based recommendation system can discover non-obvious musical similarities that analyze music beyond conventional genre classifications. By decomposing songs into latent features, we identify connections based on deeper musical characteristics rather than surface-level data.

5 Technical Implementation Details

Note: Nearly all of the mathematical functions were self-made with limited use of libraries.

5.1 Code

View the full source code

6 Acknowledgments and References

Dataset

JulianoOrlandi. "GitHub - JulianoOrlandiSpotify_Top_Songs_and_Audio_Features: GitHub, 2024, github.com/JulianoOrlandi/Spotify_Top_Songs_and_Audio_Features. Accessed 7 Mar. 2025.

Mathematical Definition References
Gene B. Kim, Stanford Stats Dept.