Harmonizing Hearts: A Music-Based Compatibility Analysis

Abhinav Acharya Tirumala Vinjamuri (B00929073) - ab806657@dal.ca
Arihant Dugar (B00917961) - ar968345@dal.ca

Objective and Scope

- Comprehensive analysis of user's music profiles.
- Similar User Matching - Collaborative & Content-Based Recommendation Algorithms.
- Song Clustering & Exploration - K-means Clustering Algorithm.
- Genre Identification - Sequential Neural Network Model.

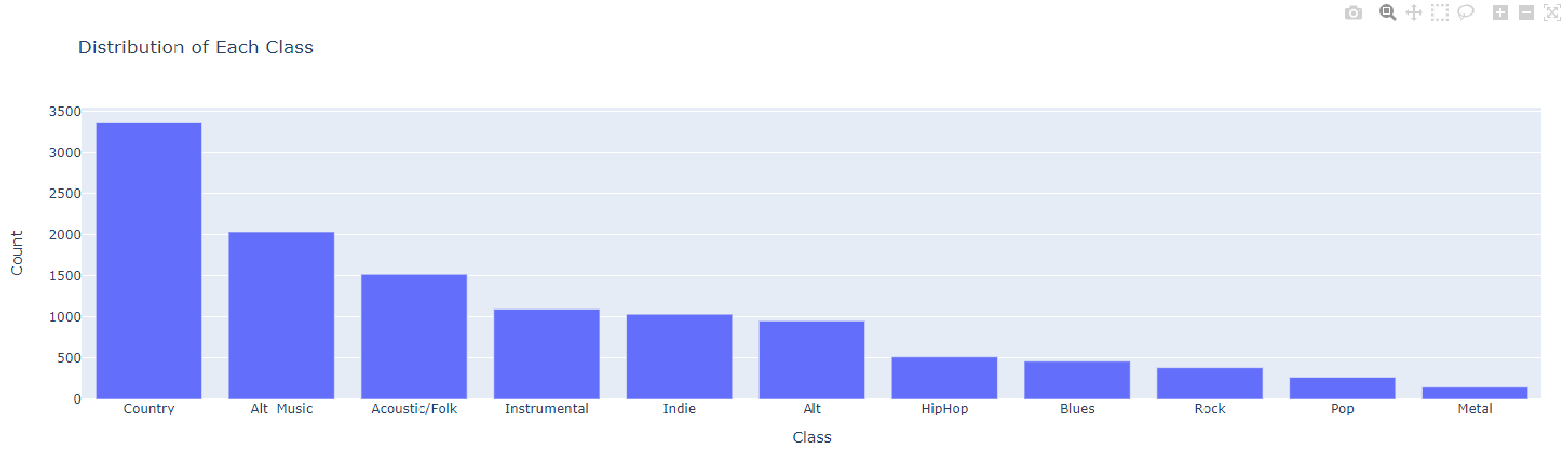
Data Collection & Preparation

Primary Dataset

- Source: Kaggle
- Original: 12,902,976 rows, 4 columns
- Limited to 250 users, 100 songs each
- Processed with Spotify API
- Final size: 28,469 rows, 19 columns

Second Dataset

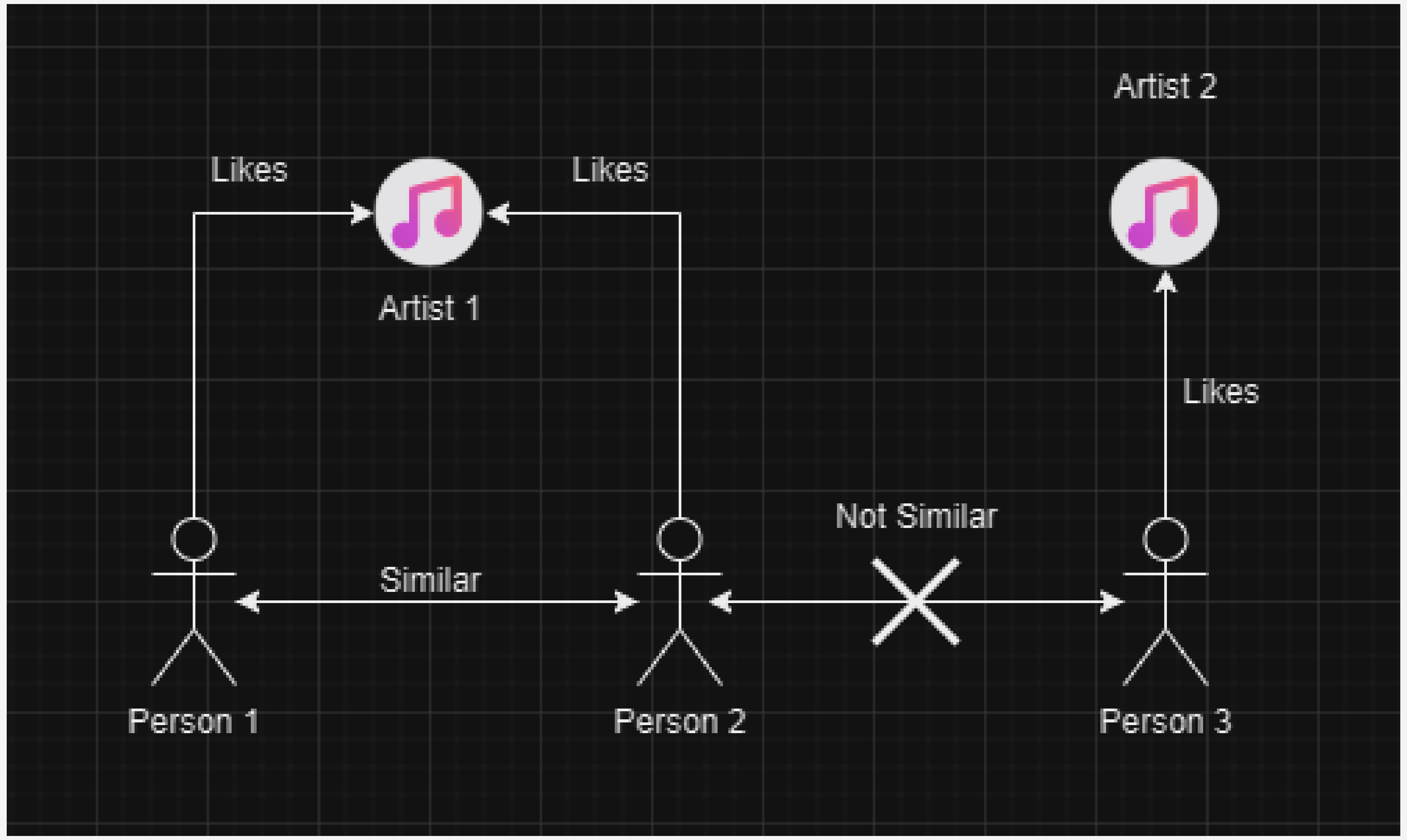
- Source: Kaggle
- Size: 17996 rows, 17 columns
- Used for Genre Identification feature
- To train and evaluate the neural network model
- 11 genres/classes



Similar User Matching - Model Explanation

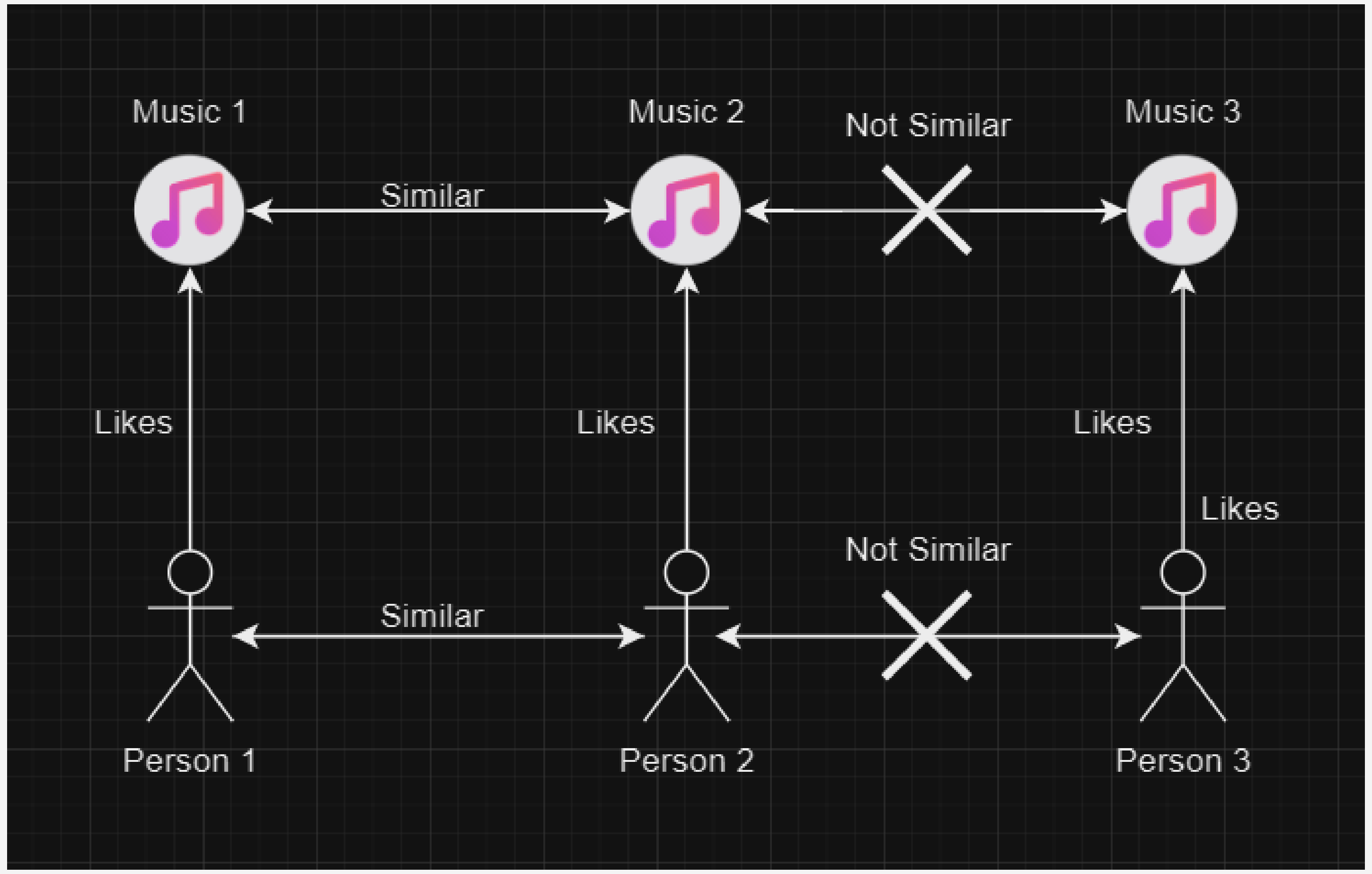
Collaborative Filtering:

- Similarity: Jaccard Similarity
- Threshold: Minimum number of common artists to be considered as relevant.



Content-Based Filtering:

- Similarity: Cosine Similarity
- Threshold: Minimum number of features that needs to be greater to be considered as relevant.



Similar User Matching

Important Components:

- User dropdown
- Specific Music Taste - Content-based algorithm
- Artists you have in common - Collaborative algorithm
- Threshold
- Evaluation Metrics
- Recommended Users

Select user

c1a6910ecac9fd5e5348326675fb6ca6

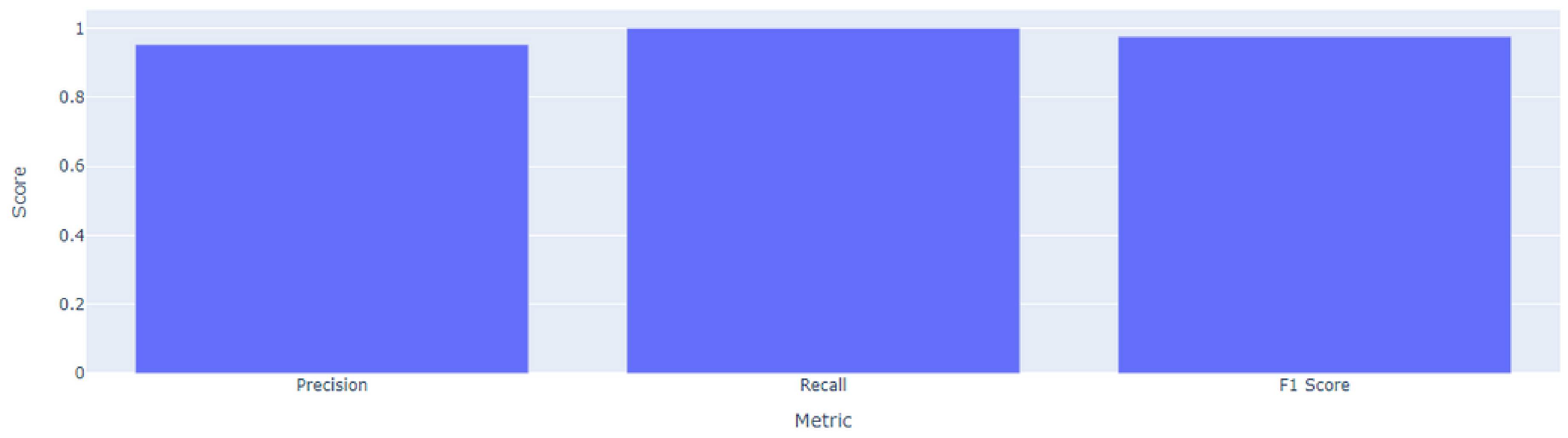
Match user according to:

Your specific music taste | Artists you have in common

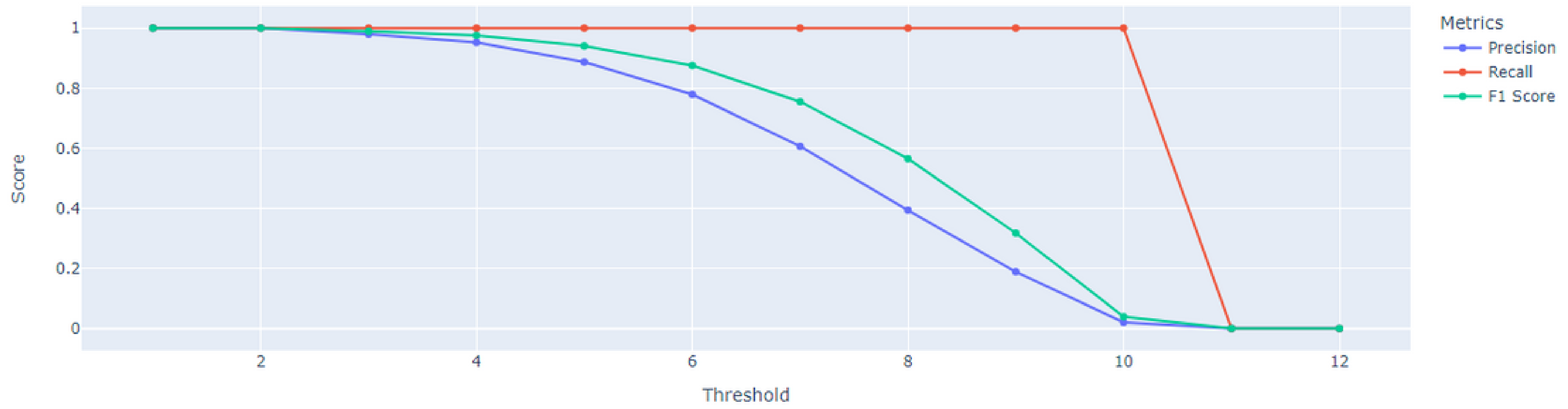
Adjust threshold (Number of features with values equal or more than your values.)

0 1 2 3 4 5 6 7 8 9 10 11 12 13

1. Evaluation Metrics



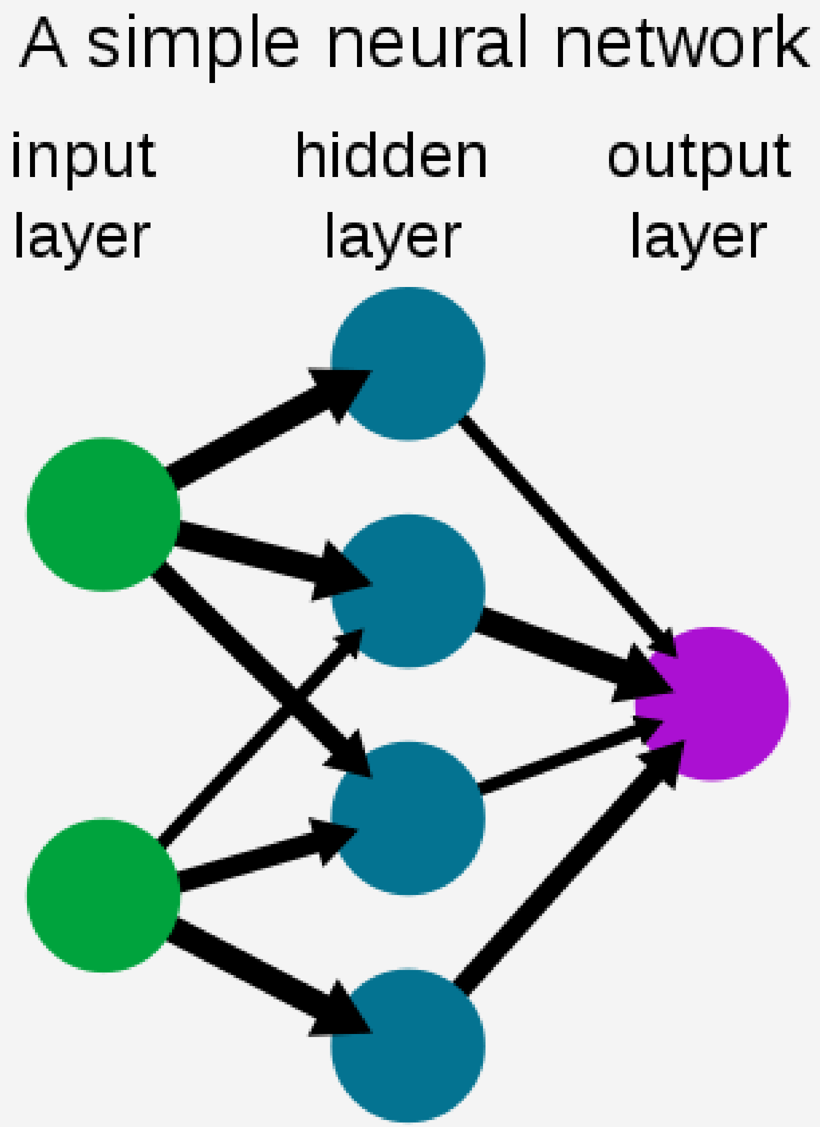
2. Evaluation Metrics at Different Thresholds



user_id_encoded	similarity	similarity_percentage
39	0.9999999999981993	99.99999999981993
183	0.9999999999821363	99.99999999821362
242	0.9999999999803794	99.99999999803794
228	0.9999999999792926	99.99999999792925
217	0.9999999999788987	99.99999999788986

Genre Identification - Model Explanation

- Model Architecture:
    - Input Layer
    - Hidden Layer
    - Dropout Layer
    - Output Layer
- Learning rate



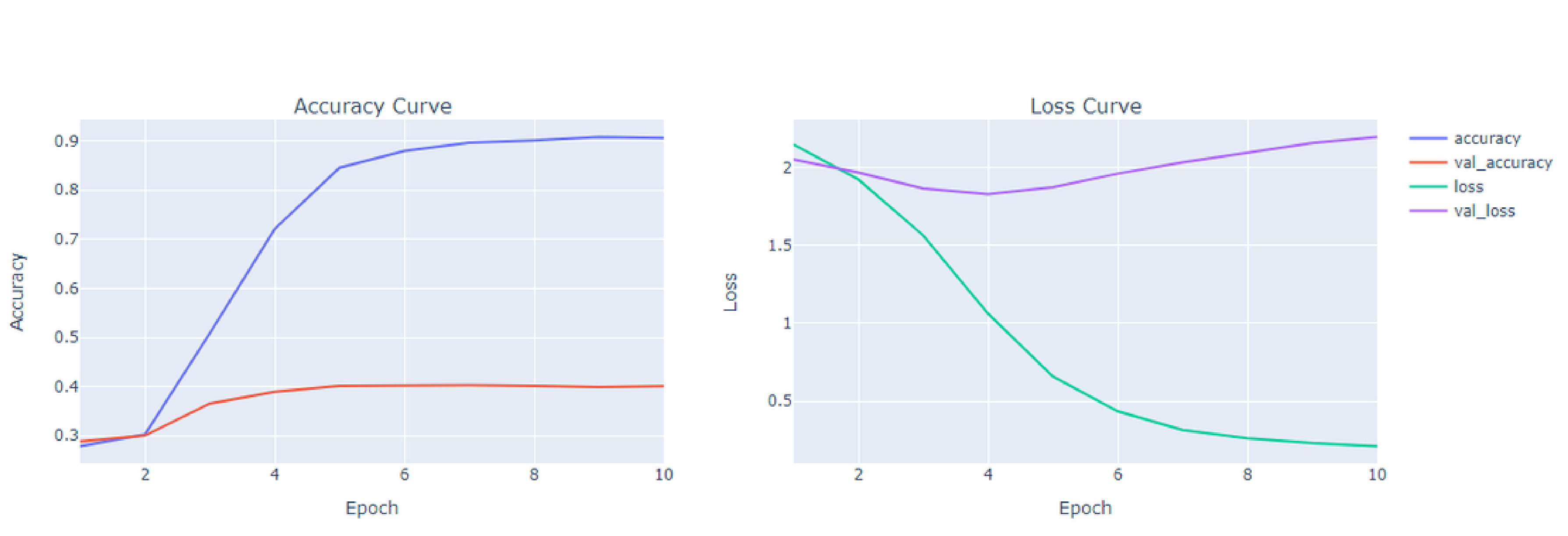
Genre Identification

Genre Identification

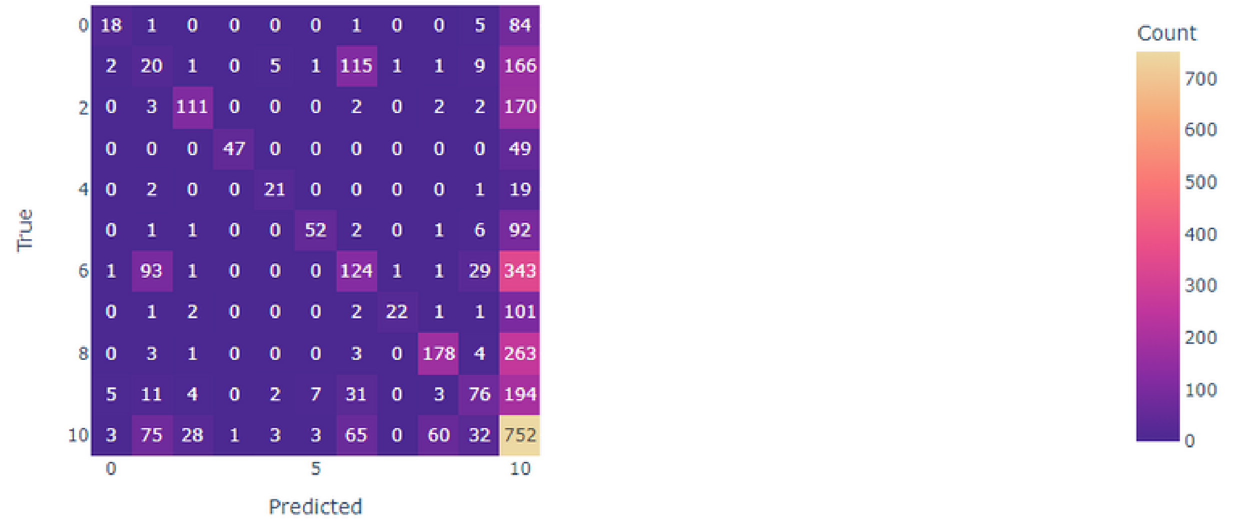
Set Learning Rate

0 0.2 0.4 0.6 0.8 1

1. Model Evaluation - Accuracy & Loss Curve



2. Model Evaluation - Confusion Matrix

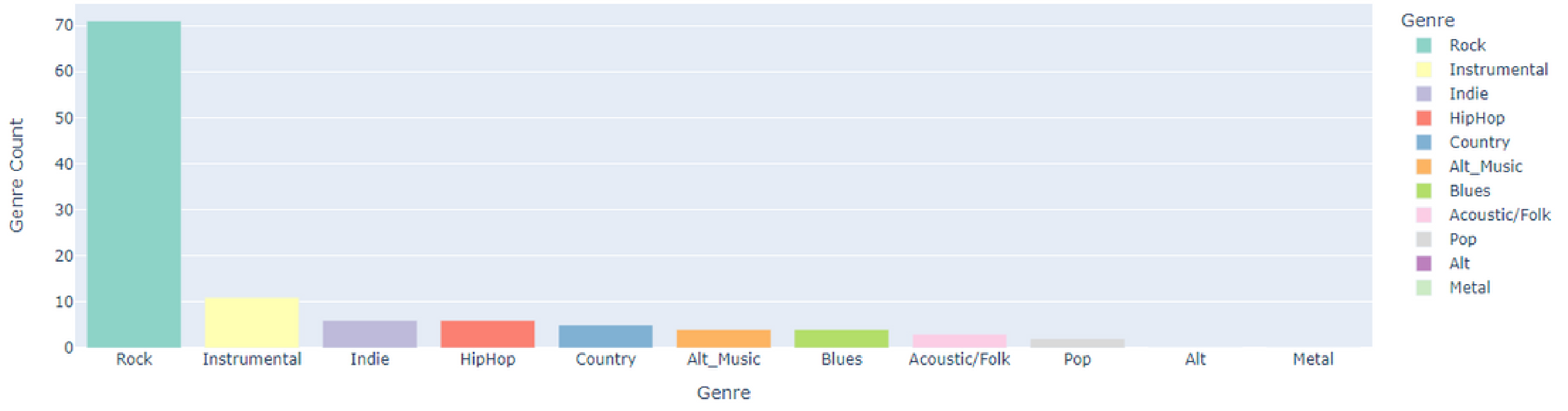


Your Data Evaluation:

Select user:

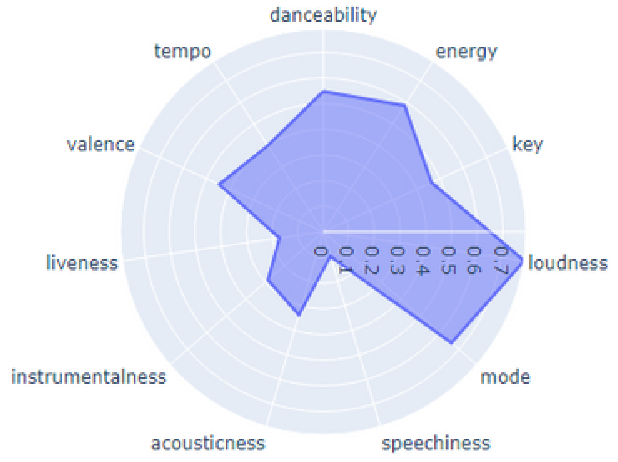
c1a6910ecac9fd5e5348326675fb6ca6

3. User Overall Genre Preference



Your most frequently listened genre: Rock

4. User Aggregated Genre Preference



Song Clustering & Exploration

Important Components:

- K-means Clustering
- Principal Component Analysis (PCA)

Song Duration Analysis

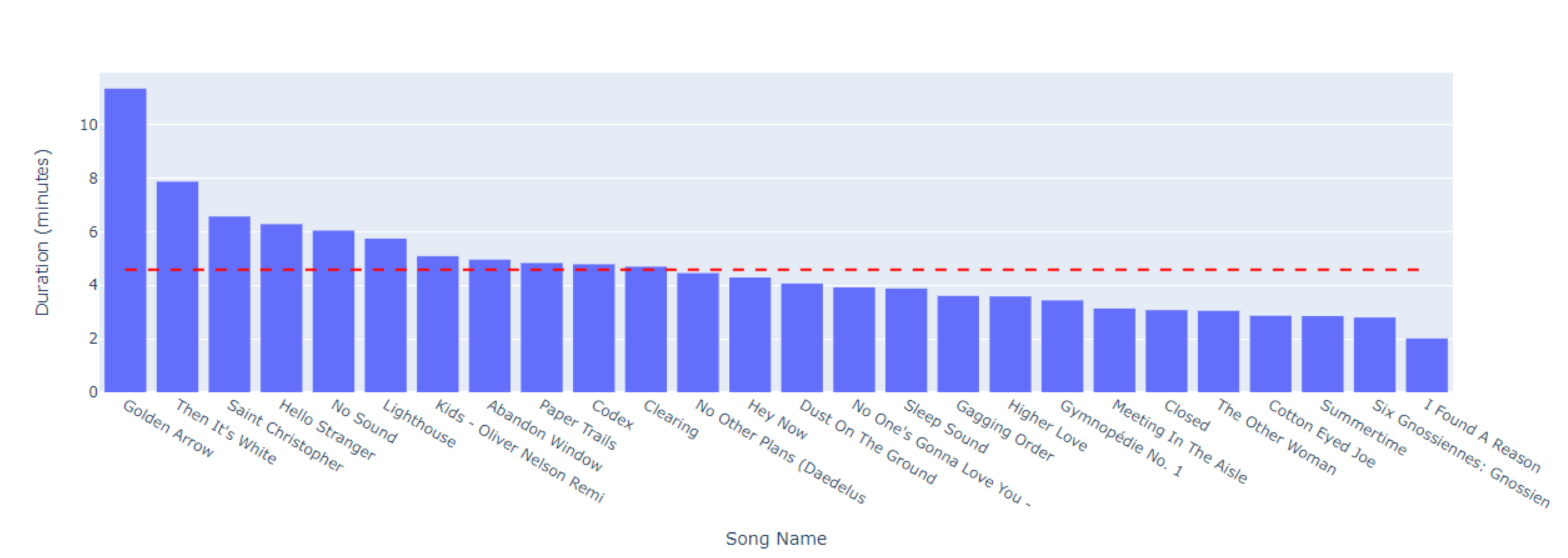
Song Duration Analysis

Select user

c1a6910ecac9fd5e5348326675fb6ca6

Select playlist

CHILL



Lesson Learned

- Music as an Emotional Tool
- User Music Selection
- Importance of Recommendation Algorithms
- Understanding Genre Identification
- Song Features for Musical Analysis
- Transparent Models in Music Analysis
- Importance of Visualization

Limitations

- Domain Challenges:
  - Limited existing work
  - Custom dataset creation
- Dataset Challenges:
  - Original Size: 4 GB
  - Reduction: Significant information loss
- Collaborative Filtering:
  - Limited Recommendations
  - Insufficient data
- Content-Based Filtering:
  - Limited Recommendations
  - Insufficient data
- Song Clustering:
  - Computational Intensity
  - Evaluation metrics
  - Challenge in slider range
- Genre Identification Challenges:
  - 11 genres in training set
  - High skewness
  - Model overfitting, unreliability

Future Works

- Song Clustering:
  - Display user position in clusters for exploration of similar songs.
- Genre Identification:
  - Formulate better datasets.
  - Optimize the model for increased reliability.
- UI Enhancement:
  - Extend project into a user-centric application.
  - Provide users access to their own data analysis.

Demo

Link: http://tinyurl.com/VA-demo

Thank You

Appendix

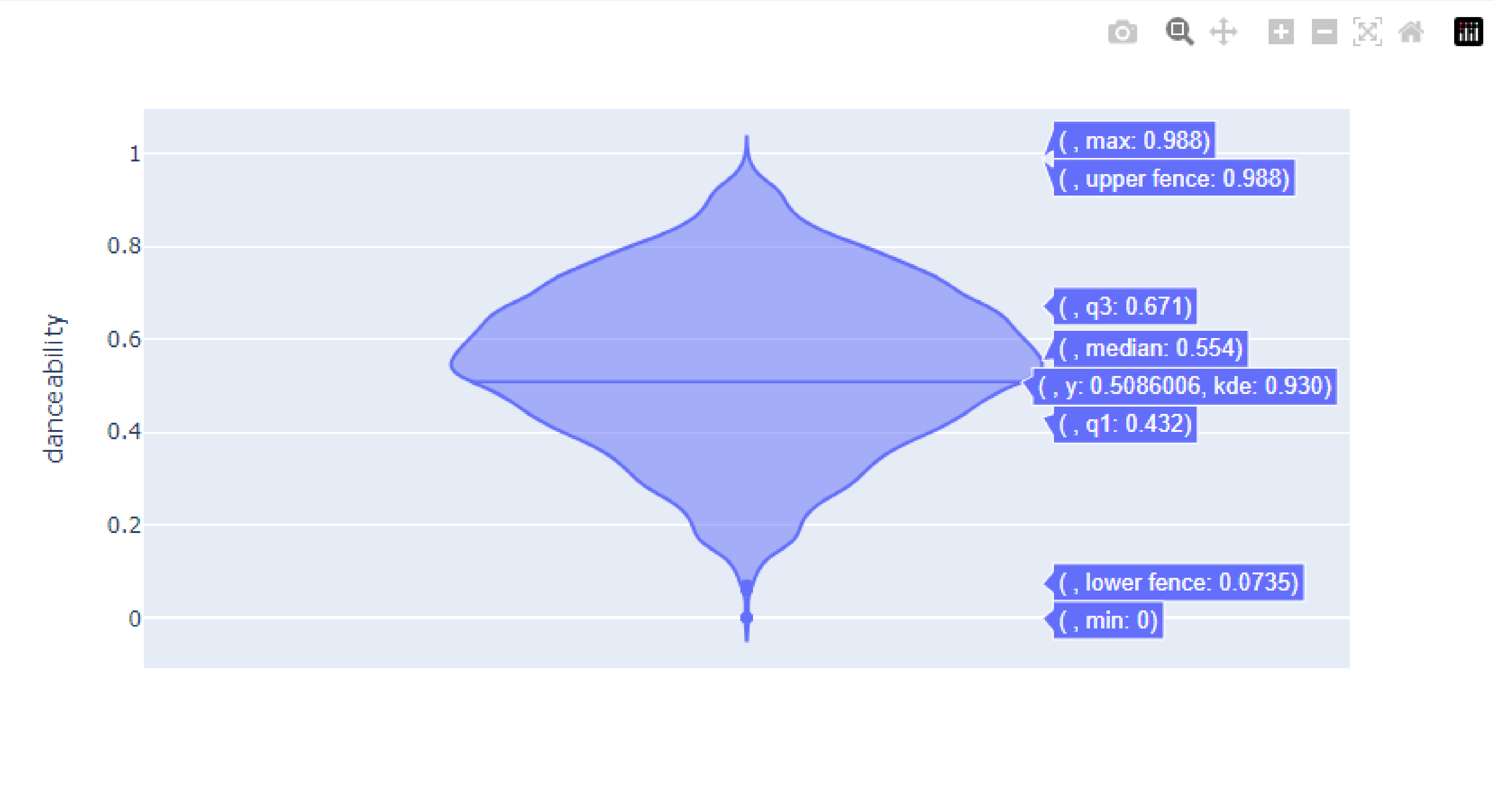
Datasources:

- Spotify Dataset: https://www.kaggle.com/datasets/abhinav1331/spotify-dataset
- Train Dataset: https://www.kaggle.com/datasets/purumalgi/music-genre-classification?select=train.csv

Appendix - Dataset

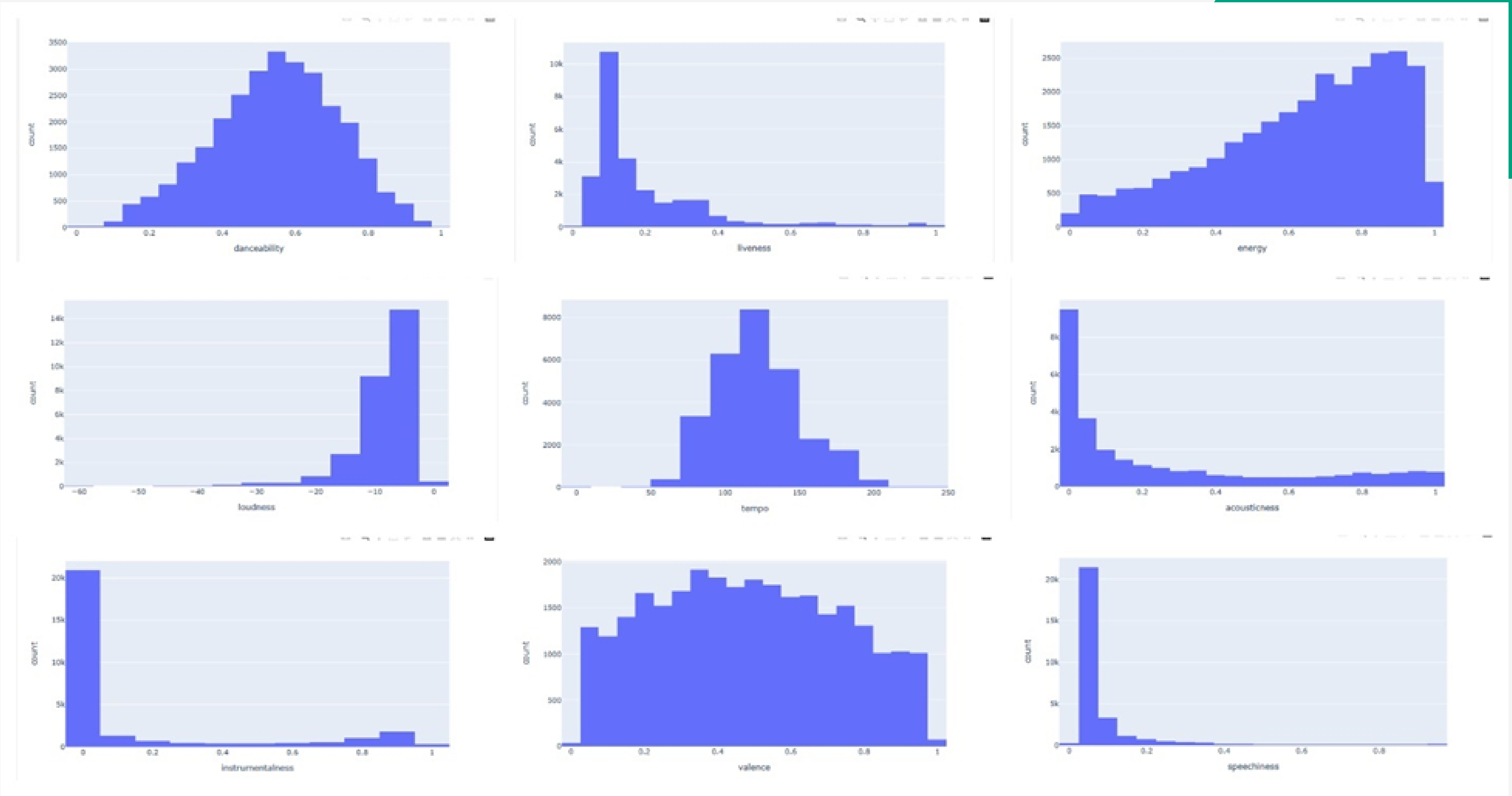
- Primary dataset source: https://www.kaggle.com/datasets/abhinav1331/spotify-dataset/data?select=spotify_data.csv.
- Secondary dataset source: https://www.kaggle.com/datasets/purumalgi/music-genre-classification?select=train.csv
- Audio Features:
  - user_id: This is the unique identifier for each user.
  - artistname: This is the name of the artist of the song.
  - trackname: This is the name of the song.
  - playlistname: This is the name of the playlist that the song belongs to.
  - song_id: This is the unique identifier for each song.
  - danceability: This is a measure of how suitable a song is for dancing based on a combination of musical elements.
  - energy: This is a measure of intensity and activity, typically perceived as fast, loud, and noisy.
  - key: This is the key the track is in.
  - loudness: This is the overall loudness of a track in decibels (dB).
  - mode: This indicates the modality (major or minor) of a track.
  - speechiness: This detects the presence of spoken words in a track.
  - acousticness: This is a confidence measure of whether the track is acoustic.
  - instrumentalness: This predicts whether a track contains no vocals.
  - liveness: This detects the presence of an audience in the recording.
  - valence: This is a measure of musical positiveness conveyed by a track.
  - tempo: This is the overall estimated tempo of a track in beats per minute (BPM).
  - type: This could be the type of the track.
  - duration_ms: This is the duration of the track in milliseconds.
  - time_signature: This is an estimated overall time signature of a track.
  - analysis_url: This could be the URL to access more detailed information about the track.
  - uri: This is the Spotify URI for the track.
  - id: This could be another unique identifier for the track or the user.

Exploratory Data Analysis - Violin Plots

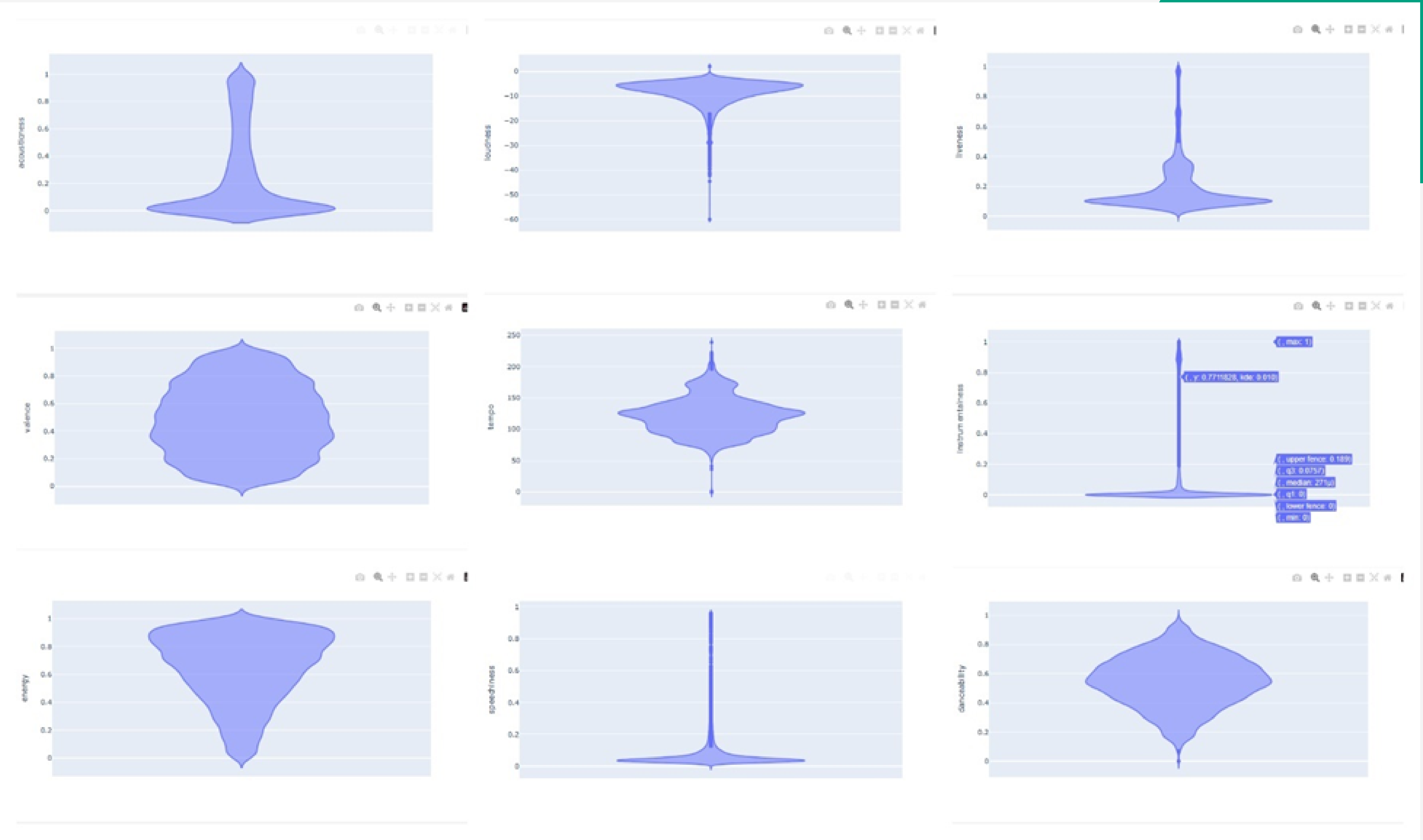


- Danceability
- Energy
- Loudness
- Speechiness
- Acousticness
- Instrumentalness
- Liveness
- Valence
- Tempo

Exploratory Data Analysis - Histogram



Exploratory Data Analysis - Violin Plots



Appendix - Tempo & Valence Analysis

- Higher tempo values imply faster-paced songs, while higher valence values indicate more positive emotions in music.
- Positive slope indicates tempo and valence grow together, whereas negative slope indicates that as one value increases, the other decreases.
- Each user had varying music taste in different playlists: Some playlists showed positive slopes while others showed negative or a straight line.

Appendix - Song Duration Analysis

- The bar plot showcases the distribution of favored song lengths, helping users understand their engagement patterns with songs of various durations, thereby allowing them to recognize their preference for specific song durations for a personalized music experience.
- Each user preferred songs that have similar length in their playlists; mostly ranging around the average length of the playlist.
- This means that users have a gullible music taste.

User Song Preference Analysis

Select user:

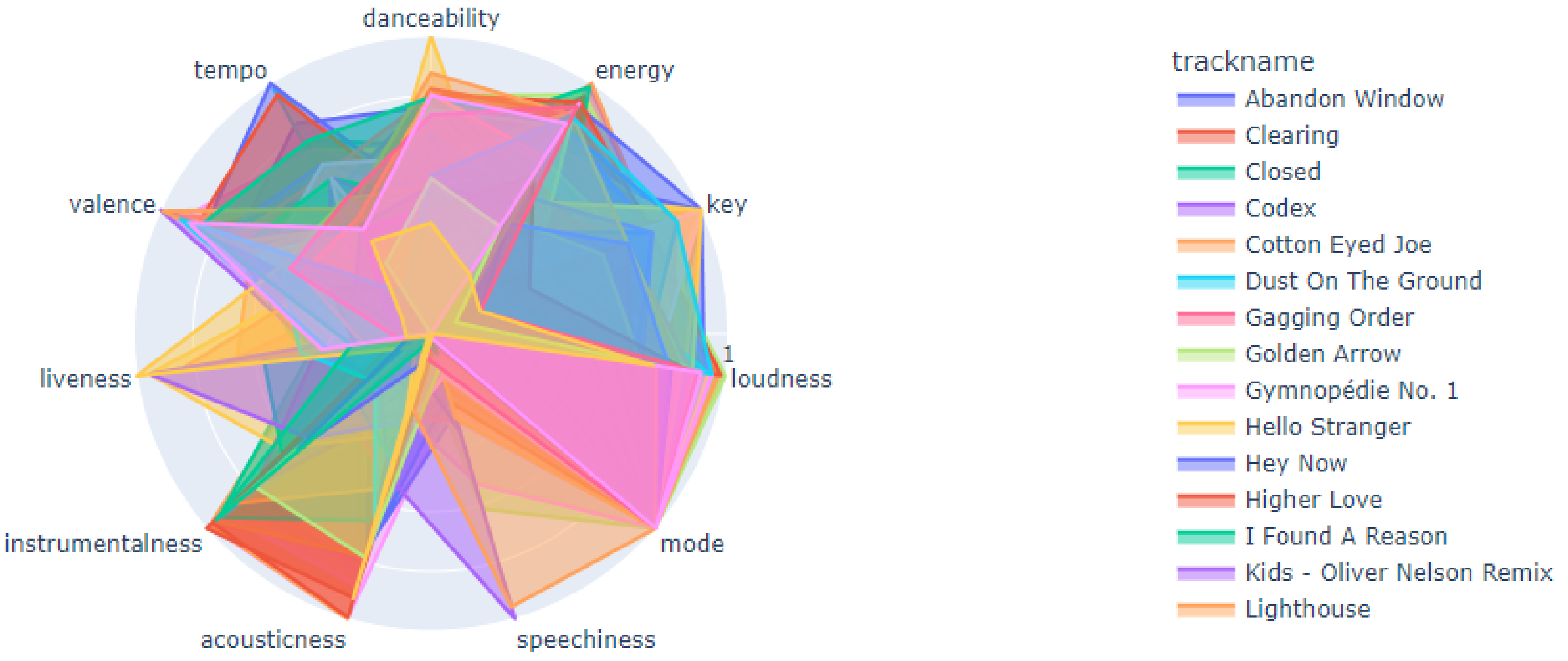
c1a6910ecac9fd5e5348326675fb6ca6

View chart for:

Average | Each song | Each playlist | Average in a playlist | Each song in a playlist

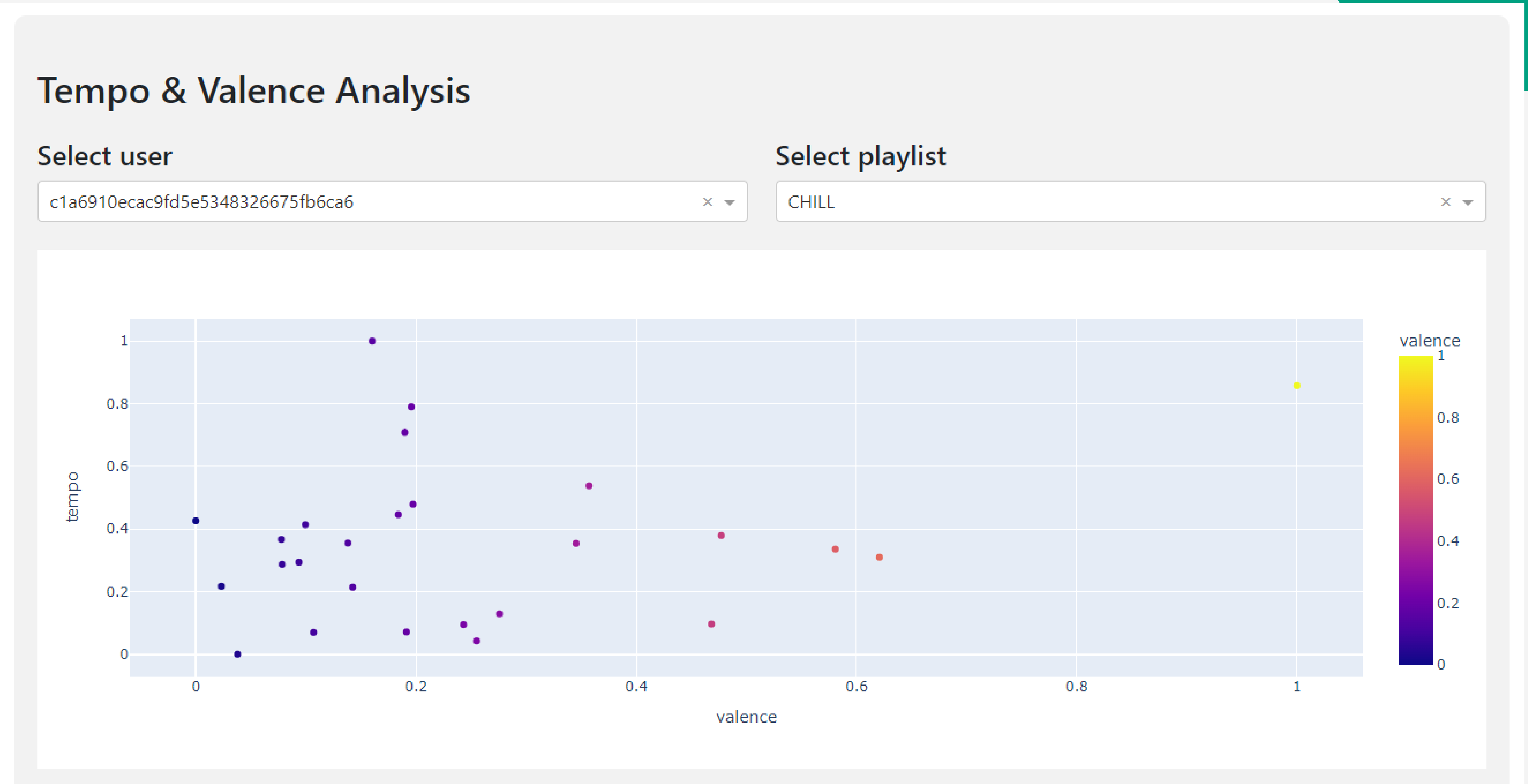
Select playlist:

default



Tempo & Valence Analysis

- Valence: Measure of musical positiveness conveyed by a track.
- Tempo: Overall estimated tempo of a track in beats per minute (BPM).



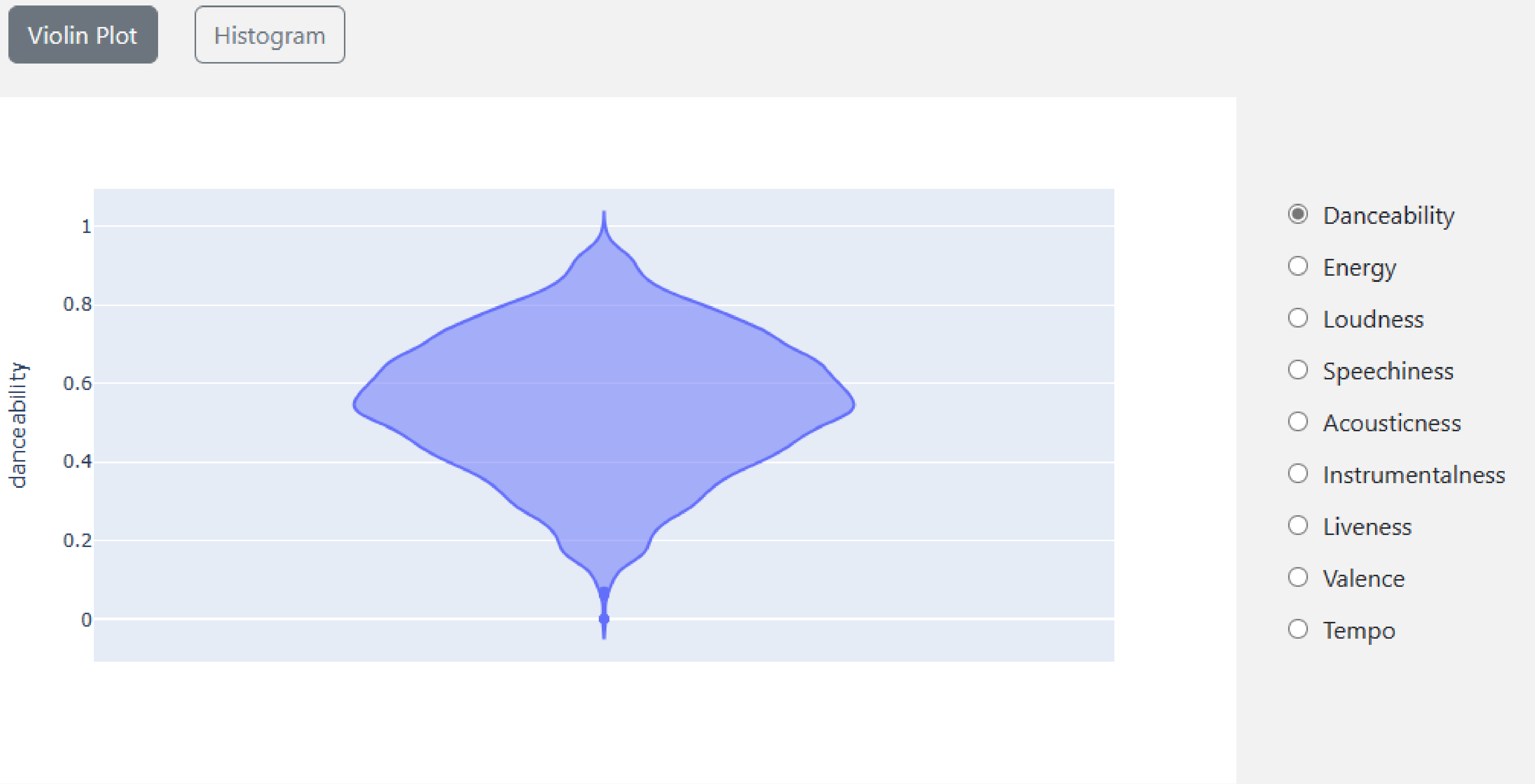
Appendix - Feature Distribution plots

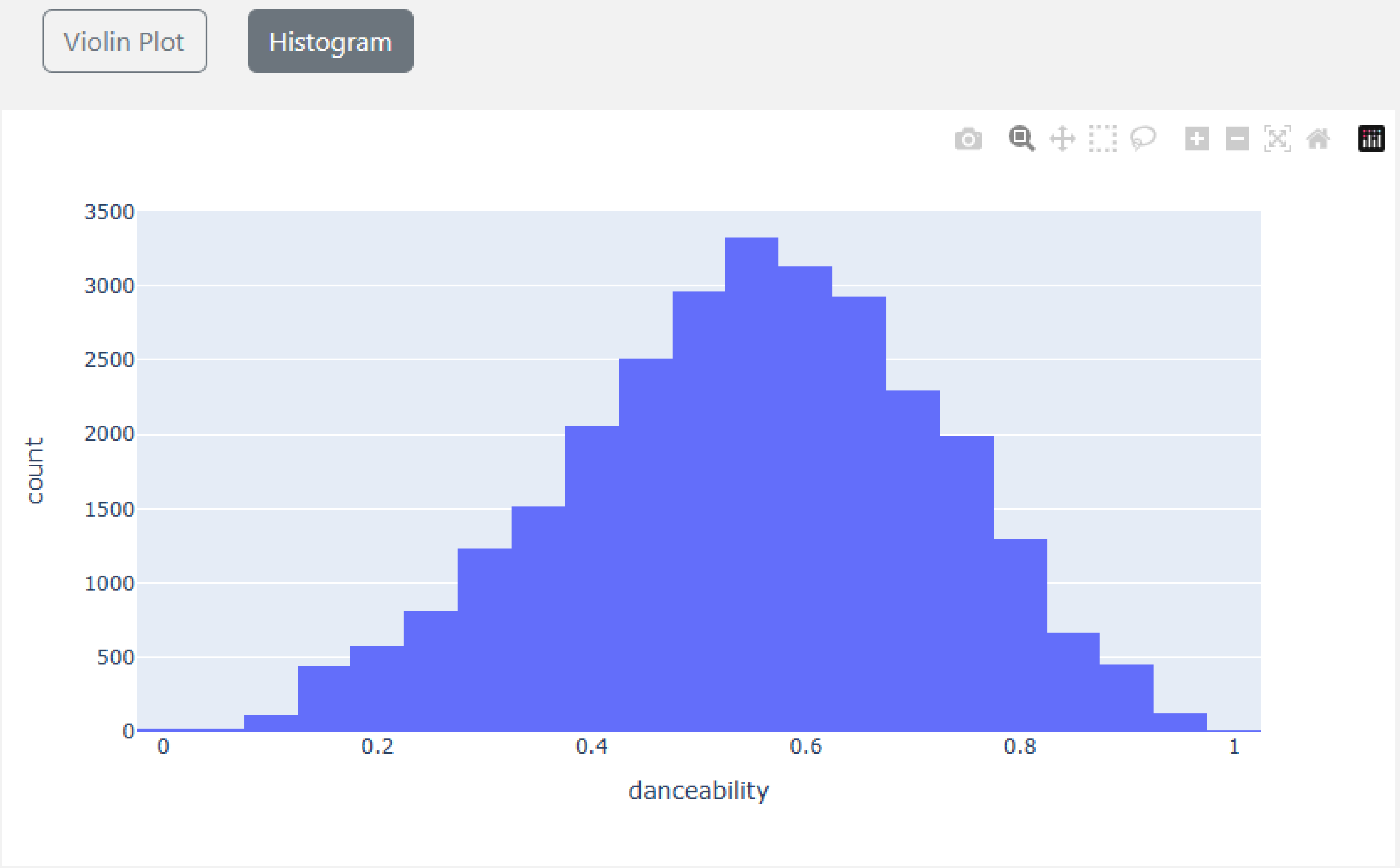
- A violin plot displays the distribution of music features like danceability, energy, and loudness, showcasing their spread and concentration. Additionally, the inclusion of a toggle button allows users to switch between violin plots and histograms, providing versatile views of feature distributions for a better understanding of their music preferences.
- Tempo, valence, energy, and danceability are normally distributed, while the other features are extremely skewed in one direction.

Appendix - Feature Correlation Heatmap

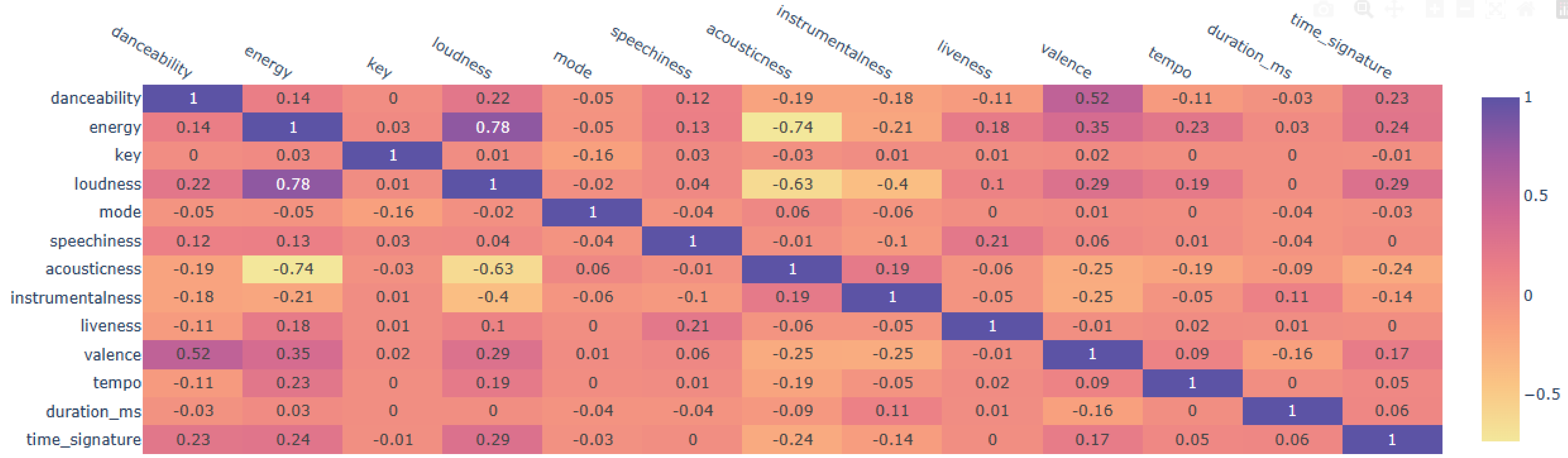
- A heatmap visually represents correlations between different features by using colors to indicate the strength of relationships. For example, if two features have a positive correlation, they will appear as a darker color, while a negative correlation will be shown as a lighter color. In the context of music features like loudness and energy, a heatmap could display how closely related these attributes are, helping users understand how these characteristics co-vary or differ in their preferred songs

Feature Distribution Plots





Feature Correlation Heatmap



Appendix - User Song Preference Analysis

- Radar charts display different song features on multiple axes, with each axis representing a specific attribute like danceability or energy. The distance from the center on each axis indicates the preference or value for that feature.
- It is evident that users liked songs that had similar feature values. This behavior can be seen for each playlist.

Appendix - Similar User Matching

- The first graph with precision, recall, and f1-score evaluates the performance of our recommendation algorithm. It measures how accurately the algorithm identifies similar users based on their music preferences.
- The second graph is a line chart displaying evaluation metrics at different thresholds. This chart illustrates the algorithm's performance concerning varying thresholds set by users. It helps users understand how changing the threshold impacts the quality and relevance of recommendations. Adjusting the threshold allows users to fine-tune recommendations and controls the models strictness.
- Collaborative Filtering:
  - Threshold: Stands for the minimum number of common artists that are required to consider a recommendation as relevant.
  - Jaccard Similarity is used as data here is 2 dimensional, each dimension being a user-artist matrix.
  - As threshold increases, the model becomes stricter in considering a recommendation as relevant.
- Content Based Filtering:
  - Threshold: Stands for the minimum number of song features that are required to be greater than or equal to the recommended user to consider a recommendation as relevant.
  - Cosine Similarity is used as data here is multidimensional, each dimension being a song feature.
  - As threshold increases, the model becomes stricter in considering a recommendation as relevant.

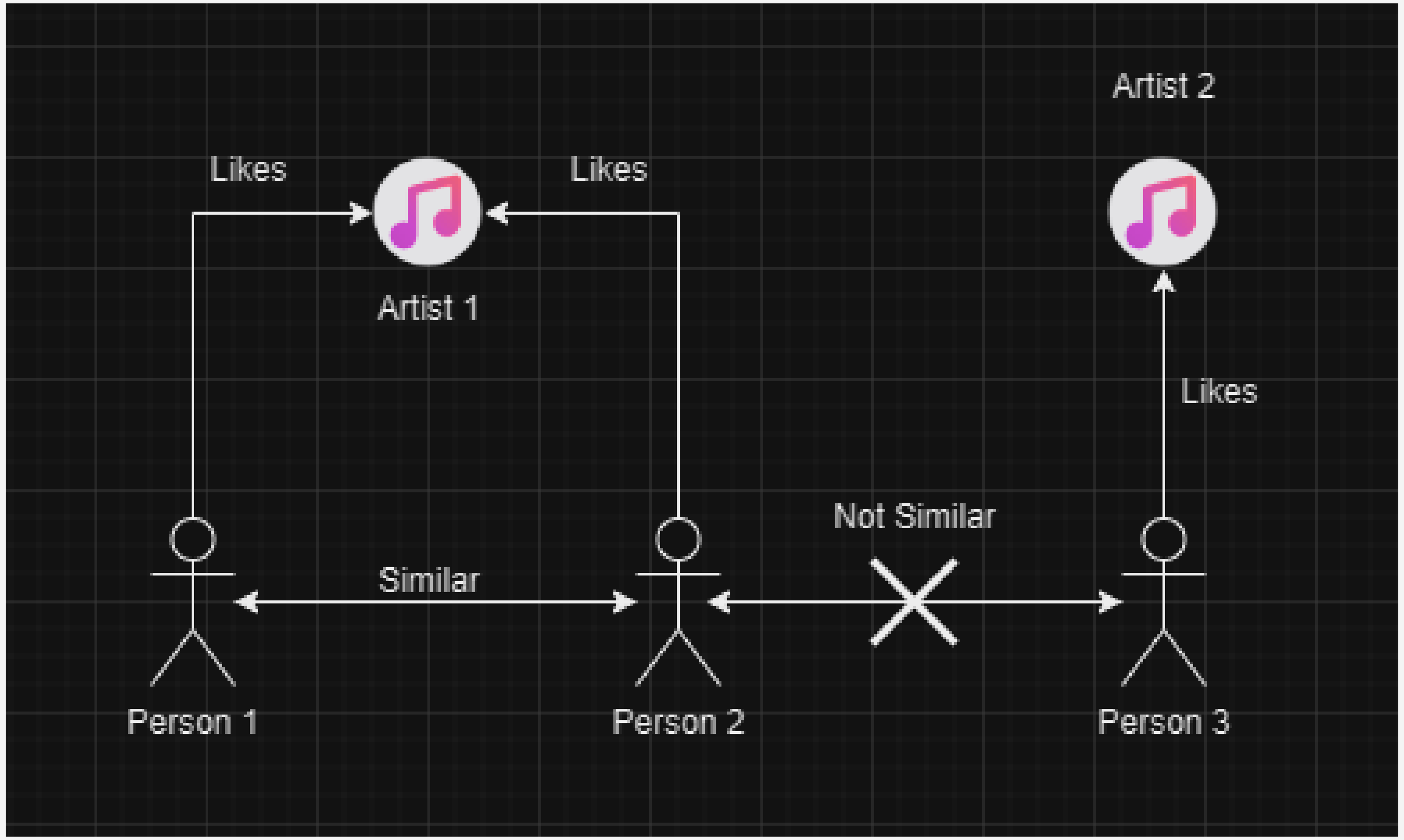
$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Jaccard Similarity

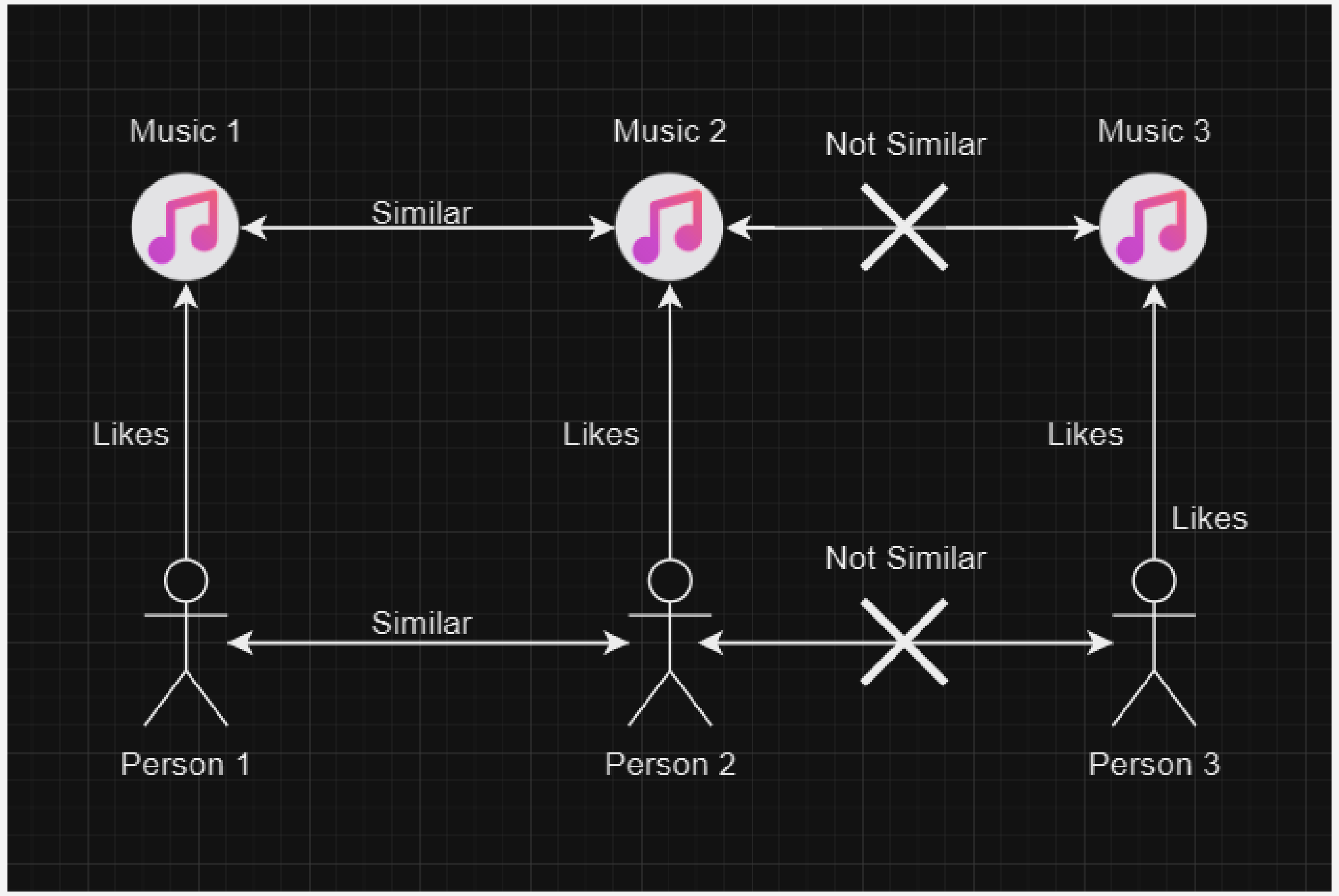
$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$$

Cosine Similarity

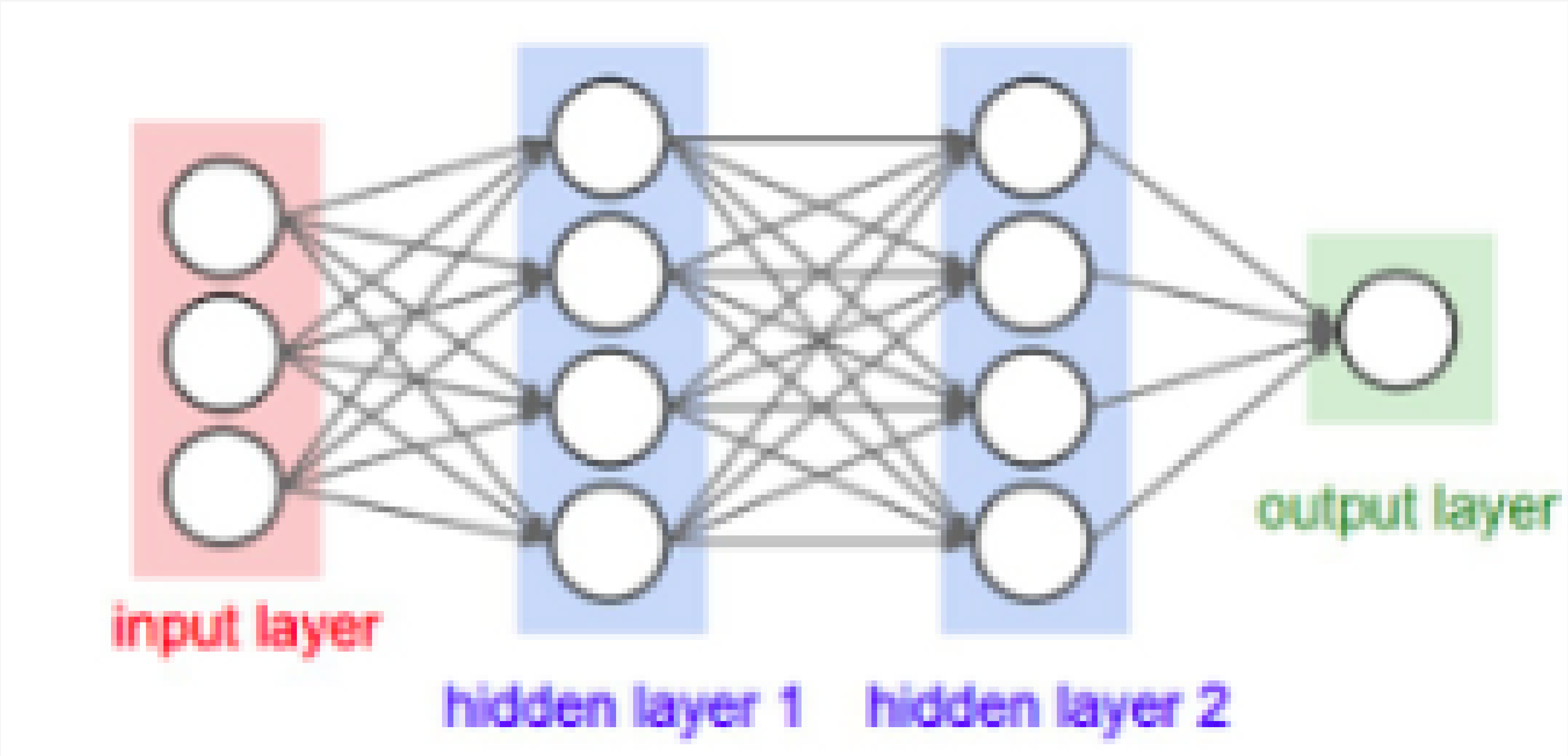
Appendix - Collaborative Filtering



Appendix - Content-Based Filtering



Appendix - Neural Network Model



Appendix - Song Clustering & Exploration

- In the scatter plot for clusters, each point represents a song. Songs with similar characteristics, like danceability and energy, are grouped together and assigned the same color or symbol, indicating their cluster. By observing the distribution of points, you can visually identify clusters or groups of songs that share common attributes. This visualization allows for the exploration of patterns and similarities among songs based on the selected features, aiding in understanding the inherent relationships between them.
- Silhouette score is a number that helps understand how well each data point fits into the cluster it's been assigned to. Higher silhouette scores mean points are well-matched to their clusters, showing better-defined clusters overall.
- We used PCA (Principal Component Analysis) to make it easier to handle the data. PCA helps by simplifying the data while keeping the important parts. This reduction in complexity makes it easier for the clustering model to work with the information and find meaningful patterns in the song's features.
- User have the ability to hover over each data point in a cluster and explore songs from that cluster.

Appendix - Genre Identification

- Model Evaluation - Accuracy & Loss Curve (line chart): This graph shows how well the machine learning model performs over time. The accuracy curve indicates how often the model correctly identifies genres, while the loss curve illustrates how the model learns and improves its predictions.
- Model Evaluation - Confusion Matrix: The confusion matrix visualizes the model`s performance by displaying the count of correct and incorrect genre predictions. It helps in understanding where the model makes mistakes and how well it classifies different genres. Ideally, we would want large values in the diagonal of the confusion matrix.
- User Overall Genre Preference: This chart presents the distribution of your most frequently listened genres. It shows the number of times each genre appears, giving an overview of users musical preferences.
- User Aggregated Genre Preference: This chart offers a comprehensive view of user's music taste across multiple genres. Each axis represents a different genre, and the plot's shape indicates user's preferences for various music genres, allowing user's to see their overall music listening pattern.
- Updating the learning rate parameter changes how the model converges and learns the underlying data patterns to classify genres. A high learning rate means that the model could underfit, whereas a low learning rate could mean that the model could overfit.