



**DALHOUSIE
UNIVERSITY**

CSCI 6612 – Visual Analytics

Project Report

Instructor: Professor Evangelos Milios.

Harmonizing Hearts: A Music-Based Compatibility Analysis

Arihant Dugar

B00917961

ar968345@dal.ca

Abhinav Acharya Tirumala Vinjamuri

B00929073

ab806657@dal.ca

ABSTRACT

"Harmonizing Hearts: A Music-Based Compatibility Analysis" addresses challenges in the music domain. Tempo and valence analysis provide valuable insights into the emotional content of music. Song duration preferences provide insights into ideal engagement length. Feature plots visualize characteristics for better understanding. User song preferences analysis enables a comprehensive view of playlist attributes, fostering a deeper understanding of personal preferences. Similar user matching recommends relevant connections. Song clustering and exploration allow users to discover songs within clusters, adding depth to musical exploration. Genre identification predicts genres and visualizes genre distributions. Each feature unravels the intricate link between music and meaningful connections.

1 INTRODUCTION

In this section, we present the main components of our project, "Harmonizing Hearts: A Music-Based Compatibility Analysis." We explain the problem we aim to solve, the importance of our work, and our proposed solution. We also discuss the novelties of our approach in terms of both visualization and machine learning. We justify and explain the machine learning techniques we used to analyze the Spotify data and the visualization modules we designed to present the results to the users. We describe the evaluation metrics we used to measure the quality and performance of our approach. Finally, we cite and acknowledge all the tools, libraries, and external resources we used in our project.

1.1 Problem Statement

"Harmonizing Hearts: A Music-Based Compatibility Analysis" aims to revolutionize online dating by addressing the intricate relationship between music and emotional connections. The problem at hand is the lack of clarity on how music influences and enhances meaningful connections in the digital dating landscape. Traditional dating platforms often overlook the role of music, resulting in a gap in understanding users' emotional preferences and connections. Our project seeks to bridge this gap, offering a unique and personalized matchmaking experience.

While one facet involves connecting users with similar musical tastes, we recognize additional issues. For instance, the lack of clarity regarding how song duration preferences influence engagement length and the need for a visual representation of users' unique music characteristics. Additionally, understanding the emotional impact of music, exploring similar songs based on the natural groups or clusters, and predicting genres based on audio features are essential components of our broader mission to offer a comprehensive and insightful analysis of a user's musical profile.

1.2 Importance of the Project

The importance of our project becomes particularly evident as we explore diverse features, each meticulously designed to delve into the intricate nuances of users' musical preferences and emotions. Whether it's the emotional journey uncovered through tempo and valence analysis, the personalized engagement achieved by considering song duration preferences, or the visual representation of individual tastes in feature distribution plots, each aspect contributes to a more personalized, meaningful, and authentic user experience. We talk more about the importance of each proposed feature below.

1.2.1 Tempo and Valence Analysis: Understanding how music influences emotions can provide valuable insights into the emotional content of music. By analyzing tempo and valence, our project seeks to understand if users prefer a unique emotional impact for different playlists.

1.2.2 Song Duration Preferences: The importance of the project further manifests in the realm of song duration preferences. Users frequently exhibit distinct preferences for the length of engagement with a song. Addressing this aspect ensures that users gain clarity on whether they tend to favor songs of similar durations more frequently. This feature adds a valuable dimension to the user's musical profile, enhancing their understanding of their preferences and contributing to a more personalized and insightful experience.

1.2.3 Feature Distribution Plots: Visualizing song characteristics such as danceability, energy, and acousticness adds another layer of importance to the project. It goes beyond conventional criteria, offering users a clear and intuitive understanding of their musical preferences. This feature enriches the platform by providing a visual map of individual taste, fostering a deeper connection between users and their music profiles.

1.2.4 Feature Correlation Heatmap: To gain a comprehensive understanding of how song features interact, it is essential to delve into their relationships with one another. The Feature Correlation Heatmap serves as a pivotal tool in unraveling these connections. By analyzing the correlations between different song features, we can gain valuable insights into the interplay and dependencies among the various attributes. This approach not only enhances our grasp of individual features but also contributes to a holistic comprehension of the intricate dynamics within a song. The Feature Correlation Heatmap stands as a cornerstone in our quest to decipher the complexities of musical attributes, guiding us toward a deeper understanding of their collective impact.

1.2.5 User Song Preferences Analysis: By comprehensively analyzing users' song preferences, we aim to provide a holistic view, enriching the user experience by understanding the intricacies of individual tastes. Diverging from traditional analysis systems that often consider only one song feature or employ dimensionality reduction, leading to vagueness, our platform stands out by considering all song features. This approach aims to transparently display individual user tastes, offering a more detailed and authentic representation of their musical preferences.

1.2.6 Similar User Matching: Understanding the profound connection between music and matters of the heart is crucial for refining the online dating experience. Music has the power to evoke emotions, memories, and shared experiences. By integrating these elements into the matchmaking process, we aim to create a more authentic and enriched platform. The project goes beyond conventional criteria like interests or hobbies, tapping into the emotional resonance of music to foster deeper connections.

1.2.7 Song Clustering & Exploration: In the realm of traditional song exploration and recommendation algorithms, there's a prevailing issue – the opaque nature of the process. These black box methods offer recommendations without providing users with insights into the underlying mechanisms. This lack of transparency results in vagueness, diminishing user trust and reliability in the recommended songs. To address this challenge, our focus is on crafting an approach that goes beyond the conventional. We aim to deliver a more explainable, easily understandable, and highly interactive method for recommending new songs. By doing so, we strive to enhance user comprehension and confidence in the songs suggested by shedding light on the intricacies of our recommendation system.

1.2.8 Genre Identification: In our project, Genre Identification plays a pivotal role by predicting genres solely based on audio features. This feature is not just about recommending genres; rather, it also focuses on accurately identifying the musical genre preferences of individual users. By doing so, it adds a crucial layer of personalization to the platform, contributing to a more comprehensive analysis. This innovative approach enhances the platform's ability to precisely understand and cater to the unique genre preferences of each user, providing a tailored and enriched musical experience.

In summary, the importance of our project lies in its ability to place the spotlight on the emotional and personal dimensions of music, where each feature within the project contributes to a more authentic, enriched, and deeply connected digital dating space. From tempo and valence analysis to user song preferences, similar user matching, song clustering and exploration, and genre identification, each facet reinforces our commitment to understanding the intricate relationship between music and meaningful connections. As users embark on this harmonious expedition, we aim to make the world of online dating a more personalized, transparent, and melodious place.

1.3 Proposed Solution

To address the intricate challenges within the music landscape, our project adopts a multi-faceted approach, encapsulated within a single comprehensive dashboard. Each feature, meticulously designed to tackle specific aspects, contributes to an enriched and personalized user experience. Here, we delve into each subsection, outlining the proposed solutions for the various facets of our project:

1.3.1 Tempo and Valence Analysis: Our solution incorporates a scatter plot on the dashboard, showcasing the correlation between tempo and valence in users' playlists. This visual representation allows users to discern the emotional content of their chosen songs, enabling a more profound understanding of the role music plays in their life in turn answering our question, whether users have a unique emotional impact for different playlists?

1.3.2 Song Duration Preferences: Recognizing users' distinct engagement preferences with song durations, our dashboard features a histogram, illustrating the distribution of preferred song lengths. This solution offers clarity on individual engagement patterns, ensuring that users can easily identify whether they tend to favor songs of similar durations. This insight contributes to a more tailored and personalized music experience.

1.3.3 Feature Distribution Plots: To provide users with a comprehensive overview of their music preferences, our feature distribution plots incorporate violin plots and histograms. These plots visualize the distribution of key features such as danceability, energy, loudness, and more. The unique addition of a toggle button allows users to switch between violin plots and histograms, offering flexibility in data presentation.

1.3.4 Feature Correlation Heatmap: Our dashboard includes a feature correlation heatmap, showcasing correlations among danceability, energy, loudness, and more. This visualization aids users in identifying patterns and relationships between their preferred features.

1.3.5 User Song Preferences Analysis: Our dashboard integrates radar charts to present a comprehensive analysis of users' song preferences. This solution goes beyond traditional systems by considering all song features, avoiding vagueness associated with one-dimensional analyses or dimensionality reduction methods. The radar charts provide users with a holistic view, deepening their understanding of the intricacies of their musical tastes. Our user song preferences analysis feature provides users with versatile options, including the ability to view average preferences, preferences for each song, preferences for each playlist, average preferences within a playlist, and preferences for each song within a playlist. This granular control empowers users to explore and analyze their music preferences at varying levels of detail.

1.3.6 Similar User Matching: Enhancing recommendations through collaborative and content-based filtering, our dashboard implements a recommendation algorithm that recommends similar users based on their preferences. This solution, visualized on the dashboard, fosters meaningful connections by

connecting users with shared musical interests, adding a layer of authenticity to the matchmaking process. The introduction of a hyperparameter called "threshold" adds a layer of relevance to the recommendations. Users can adjust the threshold, influencing the minimum number of common artists for collaborative filtering or the number of features matching for content-based filtering. This customization allows users to fine-tune the recommendation process according to their preferences.

1.3.7 Song Clustering & Exploration: Our solution enables users to explore song clusters based on selected features. Our clustering and exploration feature enhances the understanding of music preferences. Users can choose specific features such as danceability and energy to identify clusters of songs with similar characteristics. The addition of silhouette scores provides a quantitative measure of the cluster quality, aiding users in their exploration.

1.3.8 Genre Identification: Our genre identification feature utilizes a classification machine learning model to identify users' most frequently listened genre. Users can set the learning rate for the model, and the dashboard presents accuracy and loss curves, as well as a confusion matrix for model evaluation. Additionally, users can explore their personal genre identification results, accompanied by a visualization of their most frequently listened genre. The inclusion of a slider for adjusting the learning rate adds a layer of user control, allowing them to finetune the model according to their liking.

1.4 Visualization Novelties

Our music analysis dashboard stands out through its commitment to user interactivity and the introduction of novel features that redefine the exploration of musical preferences. The dashboard doesn't merely present data; it invites users into a dynamic and engaging journey of self-discovery through music.

One of the key novelties is the implementation of user-centric interactivity across all visualizations. Users can seamlessly navigate through different aspects of their music data by selecting specific users, playlists, or features. The real-time updates and dynamic adjustments in visualizations empower users to explore their musical landscape with unparalleled flexibility.

Furthermore, our visualizations offer multi-perspective insights, allowing users to view their music data from various angles. Each visualization provides different levels of detail, ensuring that users can extract meaningful insights whether they seek a broad overview or a granular examination of specific data points. An emphasis on explanatory elements adds another layer of novelty. Each visualization comes equipped with tooltips, interpretation guides, and dynamic explanations, demystifying the significance of data points and distributions. This not only enhances user understanding but also fosters a more engaging exploration of personal music preferences.

1.5 Machine Learning Novelties

Our machine learning components go beyond conventional approaches, introducing novel methodologies and user controls to enhance the predictive capabilities of the dashboard.

1.5.1 User-Controlled Hyperparameters:

- **Algorithm Customization:** Users can modify hyperparameters for all algorithms, adapting the models to their unique preferences and expectations.
- **Algorithm Transparency:** Detailed explanations are provided for each hyperparameter, empowering users to make informed decisions about model behavior.

1.5.2 Relevance in Similar User Matching:

What's novel about this approach is the enhancement it applies on relevance. Though it reduces the diversity, it enhances the relevance of the recommendations resulting in more perfect/ similar user matches which is the end goal of the task. Moreover, changing the value of the threshold as a hyperparameter gives us the control over the recommendations. As compared of traditional black box fixed recommender systems, this lets us play around with the lenience and relevance of the recommendations made. This also aids in enhancing the explainability and the interpretability of the model, as it somewhat removes the opaqueness / complexity.

1.5.3 Genre Identification with New Dataset:

- **Dataset Innovation:** A new dataset specifically curated for genre identification is introduced, allowing for more accurate and personalized genre predictions.
- **User-Driven Predictions:** The model is trained and evaluated on this new dataset, and predictions are made using user data, ensuring a tailored genre identification based on individual listening habits.

By combining cutting-edge visualizations with user-centric machine learning capabilities, our music analysis dashboard stands at the forefront of innovation, providing users with a uniquely immersive and personalized experience in exploring their musical preferences.

2 DATASET

Our project utilizes two key datasets, each contributing uniquely to the comprehensive music analysis experience offered by our dashboard.

The primary dataset, sourced from Kaggle, forms the foundation of our analysis. This dataset, an extension of an earlier dataset found [here](#), initially comprised a vast collection of 12,902,976 rows and 4 columns. Due to the sheer volume of data, we strategically limited the dataset to 250 users, each with 100 songs. Leveraging threading techniques, we processed and expanded this selection to a final size of 28,469 rows.

For each song in our refined dataset, we employed the Spotify API to extract a plethora of features, encapsulating nuanced aspects of musical content. These features include:

- **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.

To streamline our dataset, as part of preprocessing, we excluded unnecessary columns like Spotify URI and ID, ensuring a focused and relevant set of features for our project. For ease of replication of this project, we saved this dataset of 250 users with 100 songs each, with unnecessary columns removed as a separate dataset of itself and uploaded it to Kaggle (stored in spotify_data.csv).

Now in the preprocessed dataset, we converted some of the important textual columns such as user_id (hash code format), and artist_name (text) using label encoding to map each user and artist to an integer. We then, normalized the numerical columns (song-feature columns) to transform features to be on a similar scale ranging from 0-1. Its goal is to reduce redundancy and dependency within the stored information, ensuring its integrity and eliminating anomalies.

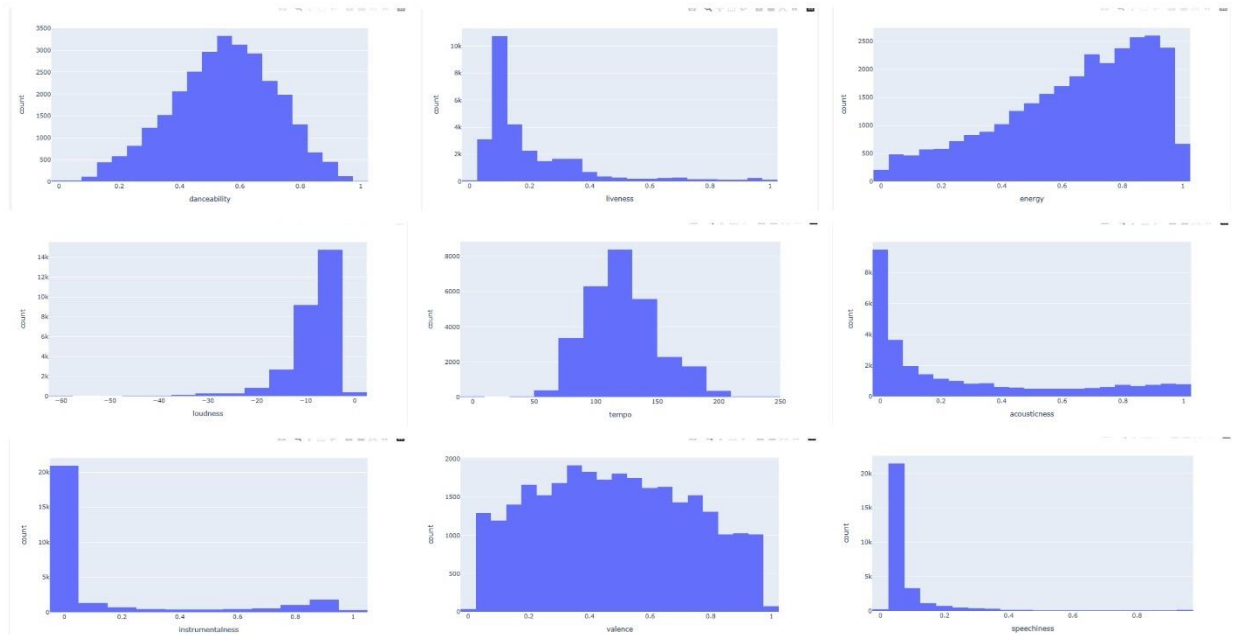


Figure 1 Distribution of song features of the primary dataset using histograms

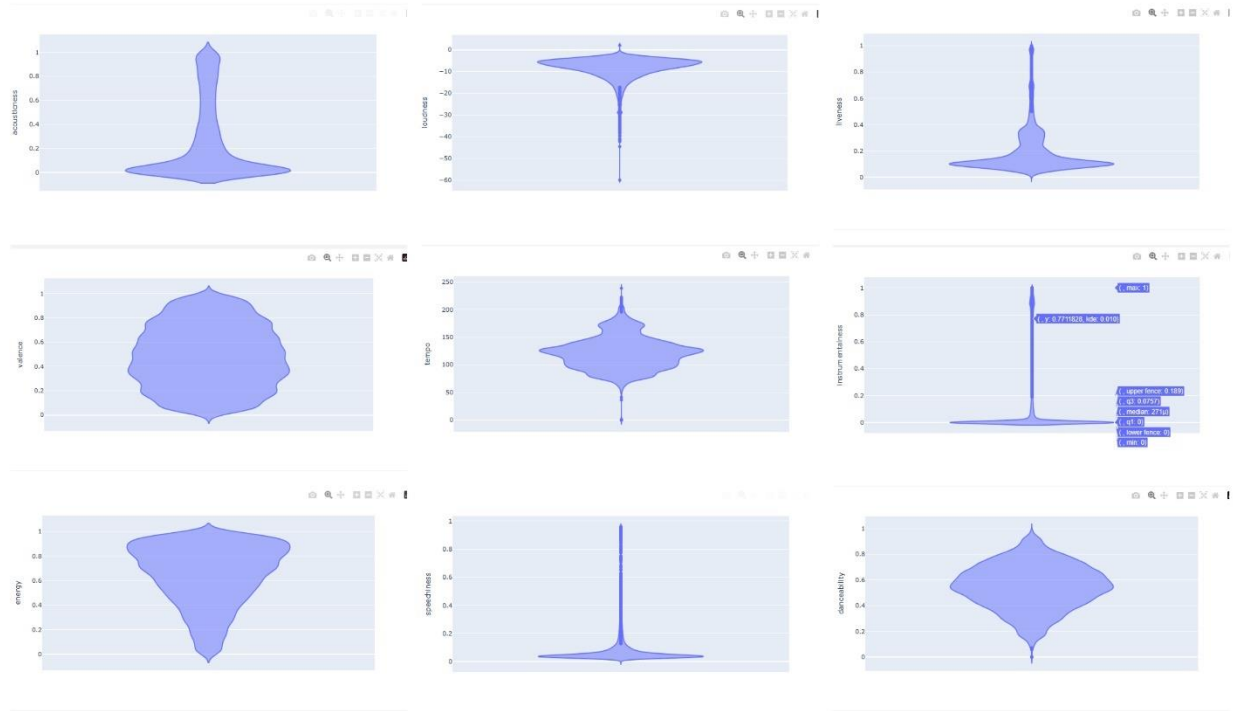


Figure 2 Distribution and Outlier detection of song features of primary dataset using violin plots.

The second dataset, crucial for our Genre Identification feature, serves a specialized purpose. Trained and evaluated on this distinct dataset (train.csv), our classification model classifies songs into 11 genres. Sourced from Kaggle, this dataset offers a curated set of instances, each associated with specific genre labels, enriching the model's ability to predict genres accurately.

As part of preprocessing, any rows containing missing values are removed to ensure data integrity, and any rows containing duplicate information is dropped. Subsequently, the columns 'Popularity', 'time_signature', and 'duration_in min/ms' are dropped from the DataFrame as they are deemed unnecessary for the analysis. The column names 'Artist Name' and 'Track Name' are then renamed to 'artistname' and 'trackname' respectively to match our primary dataset. Finally, the numerical columns, that is the song-feature columns, are normalized to a common scale for better comparability.

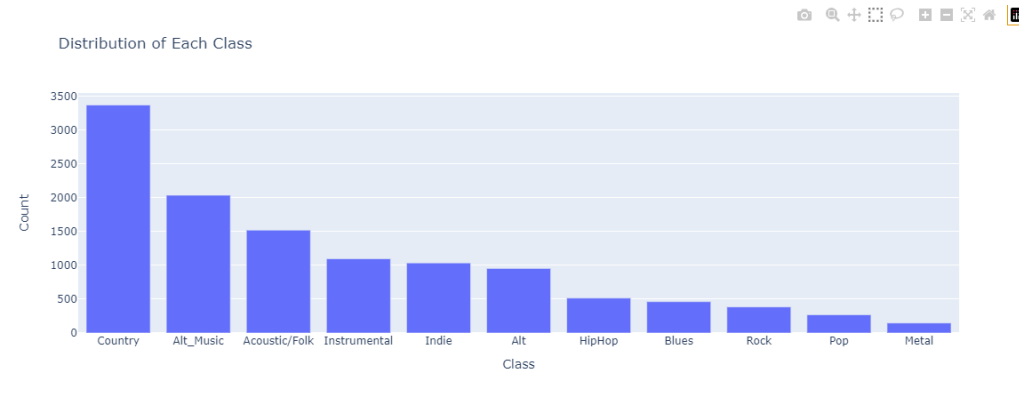


Figure 3 Genre distribution of second dataset.

Together, these datasets play a pivotal role in delivering the diverse and personalized insights encapsulated within our music analysis dashboard. They enable us to provide users with a rich, data-driven exploration of their music preferences.

3 MACHINE LEARNING MODEL EXPLANATION

In this section, we will go over each algorithm that we implemented in the project. Mainly, we will go over the objective, the general idea of how the model works, and the implementation details.

3.1 Similar User Matching

Objective: Enhance music recommendations by identifying users with similar musical preferences. To achieve this functionality, we employed a subset of recommendation algorithm family: collaborative and content-based filtering models.

Justification: The selection of collaborative and content-based filtering recommendation algorithms for improving music recommendations is driven by the unique characteristics of the task. Collaborative filtering draws on user interactions to create meaningful connections based on shared artists, while content-based filtering focuses on personalized song feature preferences, offering fine-grained recommendations. These algorithms are adept at handling continuous and multidimensional data inherent in music preferences. Their efficiency with large datasets ensures scalability, making them well-suited for diverse musical tastes. In essence, the adoption of collaborative and content-based filtering aligns strategically with the continuous and multidimensional nature of music preferences, enhancing matchmaking with nuanced and accurate recommendations.

The implementation specifics are explained in the sections below:

3.1.1 Collaborative Filtering

Objective: Enhance music recommendations by identifying users with similar musical preferences in terms of artists listened to.

Idea: Collaborative Filtering leverages the preferences and behaviors of a group of users to make recommendations. The underlying principle is that users who agreed in the past tend to agree again in the future.

Implementation Details: In our implementation, we use a Collaborative Filtering approach with a twist. We calculate the similarity between users using Jaccard similarity, a metric well-suited for two-dimensional data.

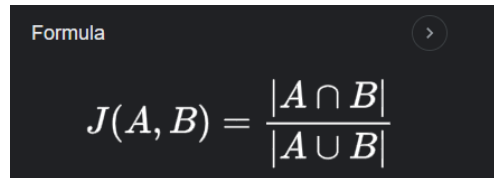

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Figure 4 Jaccard Similarity formula

In the context of collaborative filtering, these two dimensions (A, B) represent users. The User-Artist Matrix is created, where each cell contains the interaction of a user with a particular artist. The similarity between users is calculated using Jaccard similarity, and recommendations are made and returned in a descending order of similarity percentage: most similar users at the top, least similar users at the bottom. The model calculates the similarity score for 249 users out of our 250-user dataset. As a result, the model returns a list of 249 users with different similarity scores.

To add a layer of control for users, we introduced a hyperparameter, the threshold. This threshold determines the minimum number of common artists two users must have to be considered a relevant recommendation. This hyperparameter allows users to tailor the strictness of the recommendation algorithm. As the threshold value increases, the algorithm gets stricter and its precision increases, whereas its recall and f1-score decreases. To put it in a simpler way, different values of threshold consider different number of users out of the 249 recommended users as relevant. More on this hyperparameter and how it works will be discussed in the model evaluation section. Ideally, the users must select a threshold value that results in high values of f1-score, and precision. Together, the similarity and the relevance factor form a unique novel approach to recommendation algorithms that sets us apart from the existing implementations.

3.1.2 Content-Based Filtering

Objective: Enhance music recommendations by identifying users with similar song feature preferences.

Idea: Content-Based Filtering recommends items based on the features of the items and a profile of the user's preferences. In the context of our project, it involves recommending users with similar song feature preferences.

Implementation Details: To capture the multidimensional nature of song features, we opt for cosine similarity, a suitable metric for comparing vectors. Features like danceability, energy, etc., are used, and each user is assigned a profile based on their song preferences. The profile is nothing, but an aggregation of song features for all the user's songs in the dataset. The similarity between users is then computed based on the cosine similarity of their feature preferences.

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

Figure 5 Cosine similarity formula

A, and B in this implementation stands for the vectors of 2 users, while the vectors are the users' music profiles, i.e. aggregated song-feature values for all their songs in the dataset. This approach empowers users to connect with others who share similar tastes in song features, which is far more refined than the collaborative filtering method. The model calculates the similarity score for 249 users out of our 250-user dataset. As a result, the model returns a list of 249 users with different similarity scores in a descending order of similarity percentage: most similar users at the top, least similar users at the bottom.

As an additional layer of control for users, we introduced a hyperparameter called the threshold. This threshold determines the number of songs features of the recommended user that need to be equal or greater than the features of user that is being recommended to, to be considered as a relevant recommendation. As the threshold value increases, the algorithm gets stricter and its precision decreases, whereas its recall and f1-score decreases. To put it in a simpler way, different values of threshold consider different number of users out of the 249 recommended users as relevant. More on this hyperparameter and how it works will be discussed in the model evaluation section. Ideally, the users must select a threshold value that results in high values of f1-score, and precision. Together, the similarity and the relevance factor form a unique novel approach to recommendation algorithms that sets us apart from the existing implementations.

3.2 Song Clustering & Exploration

Objective: Uncover patterns and relationships within the music dataset through clustering, enabling users to explore and understand the inherent structures.

Justification: To achieve this, we employed a clustering model that categorizes songs based on their feature similarities, facilitating exploration and analysis. Unlike traditional genre categorization, K-Means operates in a high-dimensional feature space, allowing it to identify complex patterns and relationships among songs. This algorithm excels in creating clusters based on multiple musical attributes, providing users with a sophisticated tool to navigate their music collection. By leveraging the K-Means algorithm, users can uncover latent structures within their songs, enabling a personalized exploration experience. It's not just about clustering; it's a machine learning-driven exploration of music, unraveling intricate patterns and connections that redefine the way users engage with their music library.

The details of the model and its implementation are outlined below:

3.2.1 K-Means Clustering

Objective: Group songs into distinct clusters based on their feature similarities.

Idea: K-Means Clustering partitions the dataset into k clusters, with each cluster center representing a group of similar songs. The algorithm iteratively refines the cluster centers until convergence.

Implementation Details: We chose K-Means as our clustering algorithm for its simplicity and effectiveness. Users can select the song features for clustering, such as danceability, energy, etc. The model then groups the songs into clusters, allowing users to explore songs that share common characteristics.

The silhouette score, a metric indicating how well-separated the clusters are, is calculated for different values of k (number of clusters). Users can visualize the silhouette scores to determine the optimal number of clusters for their exploration needs. To enhance user control, we expose hyperparameters like the number of clusters (k) and song features for clustering. Users can fine-tune these parameters to obtain clusters that align with their preferences and analytical goals.

Additionally, the exploration facet provides users with an interactive 3D scatter plot, where each point represents a song and is colored based on its cluster assignment. This visual representation empowers users to visually identify patterns and explore relationships between songs within and across clusters. This combination of user-configurable parameters, silhouette score evaluation, and interactive visualization contributes to a dynamic and user-friendly song clustering and exploration feature. This approach enables users to derive meaningful insights from the dataset while maintaining a high level of control over the exploration process.

3.3 Genre Identification

Objective: To recognize patterns in song features and accurately predict the genre.

Justification: To achieve this feature, we implemented a neural network model which is part of the classification algorithm family from the broader realm of deep learning. After comparing multiple classification algorithms such as random forest, decision tree, naïve bayes, etc., it was evident that neural networks model provided the best performance for the problem at hand.

3.3.1 Neural Network Model.

Idea: Neural networks, inspired by the human brain, consist of interconnected nodes organized in layers. The model learns complex patterns by adjusting weights during training, enabling it to make genre predictions based on input features.

Implementation Details: We designed a neural network with multiple layers, including input, hidden, and output layers. For training, we utilized a dedicated dataset (train.csv) with 11 genre classes. The model underwent an iterative process of forward and backward passes, adjusting weights through backpropagation to minimize prediction errors. The training process involves user-provided control over hyperparameters, such as the learning rate. Users can fine-tune this parameter to regulate the model's adaptation during training.

The neural network architecture comprises several layers, each serving a specific purpose in the learning process. Here's a more detailed breakdown:

Input Layer: Nodes in the input layer correspond to different song features, including danceability, energy, and more.

Hidden Layers: Multiple hidden layers facilitate the learning of intricate patterns and relationships within the input data.

Activation functions, specifically Rectified Linear Unit (ReLU), are applied to the nodes in the hidden layers. ReLU introduces non-linearity to the model, allowing it to capture complex mappings between inputs and outputs. This is crucial for the neural network to learn and represent more intricate patterns in the data.

Output Layer: The output layer contains nodes representing different genre classes (e.g., Pop, Rock, Jazz). A softmax activation function is employed in the output layer. Softmax converts the raw output values into probabilities, enabling the model to make a confident prediction about the genre of a song. This is particularly useful for multi-class classification tasks.

Loss Function: The categorical crossentropy loss function is utilized. Categorical crossentropy is well-suited for multi-class classification problems, penalizing the model more when its prediction diverges from the actual class. Minimizing this loss during training ensures the model converges towards accurate genre predictions.

Optimizer: The Adam optimizer is employed to adjust the weights of the neural network during training. Adam combines the advantages of two other popular optimizers, AdaGrad and RMSProp, offering efficient adaptation to different learning rates and improved convergence.

Learning Rate: Users have the flexibility to control the learning rate as a hyperparameter. The learning rate determines the step size in weight adjustments during training. A higher learning rate may speed up convergence but risks overshooting optimal weights, while a lower learning rate may ensure stability but slow convergence. Allowing users to modify this parameter provides a tailored training experience.

In summary, the neural network model for genre identification incorporates ReLU activation functions for non-linearity, softmax activation in the output layer for probabilistic predictions, categorical crossentropy loss for multi-class classification, and the Adam optimizer for efficient weight adjustments during training. The user-centric control over the learning rate enhances adaptability, enabling users to fine-tune the model's performance based on their preferences and data characteristics.

4 VIZUALIZATIONS EXPLANATION

Tempo & Valence Analysis: This feature aims to help users discern the emotional content of their selected songs by exploring the correlation between tempo (speed) and valence (emotional positivity/negativity). To achieve this, a scatter plot was chosen, as it effectively displays the relationship between two continuous variables. The visualization allows users to understand whether there exists a correlation between song speed and emotional tone. Users can quickly grasp whether they have a pattern of emotional impact for different playlists.

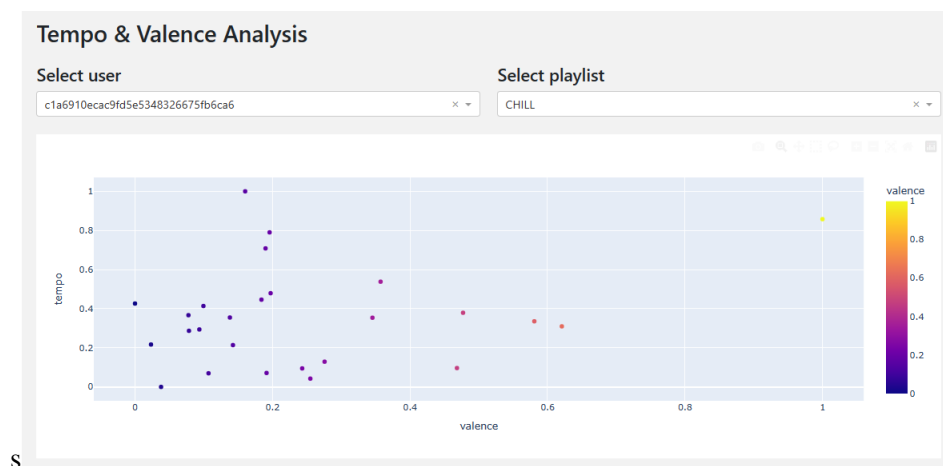


Figure 6 Tempo vs Valence Analysis on dashboard

Song Duration Analysis: The purpose of this feature is to showcase the distribution of song durations within the user's playlists or selections. A histogram was chosen for its effectiveness in displaying distributions and frequencies of continuous data. This visualization provides insights into the spread and frequency of different song durations, enabling users to understand the prevalent length of songs in their collections.

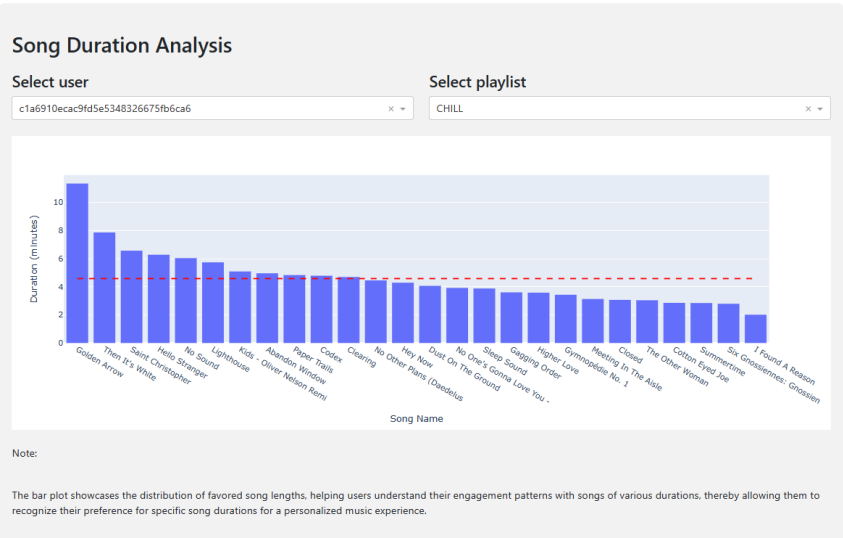


Figure 7 Song Duration Analysis on dashboard

Feature Distribution Plots: This feature utilizes violin plots and histograms to effectively demonstrate the distribution of various song features across playlists or users. Violin plots are used over the traditional box plots as they not only identify the outliers, but also display the spread of the data. Histograms are a best way of portraying the spread and distribution of data. The chosen visualizations illustrate the diversity and variations in song characteristics like danceability, energy, loudness, etc. Users can comprehend how different features are spread among their playlists or selections, gaining insights into the range and diversity of these attributes.

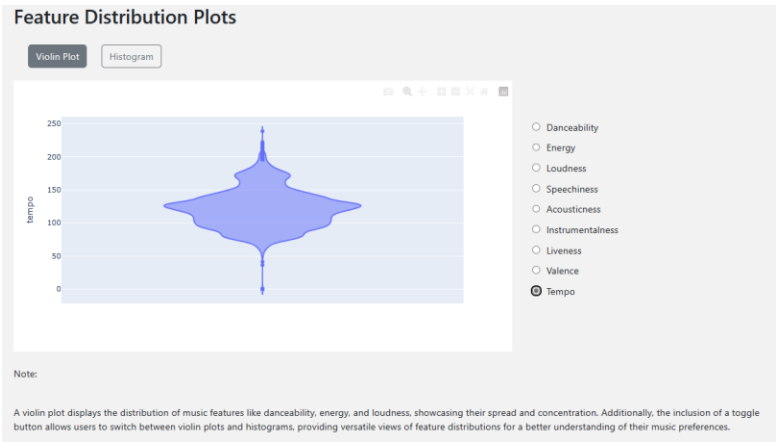


Figure 8 Feature Distribution plots on dashboard

Feature Correlation Heatmap: For visualizing relationships between different song features, a heatmap was chosen. Heatmaps are excellent for displaying correlations between multiple variables. The

visualization offers insights into how song characteristics are interrelated, enabling users to identify which attributes tend to correlate with each other.

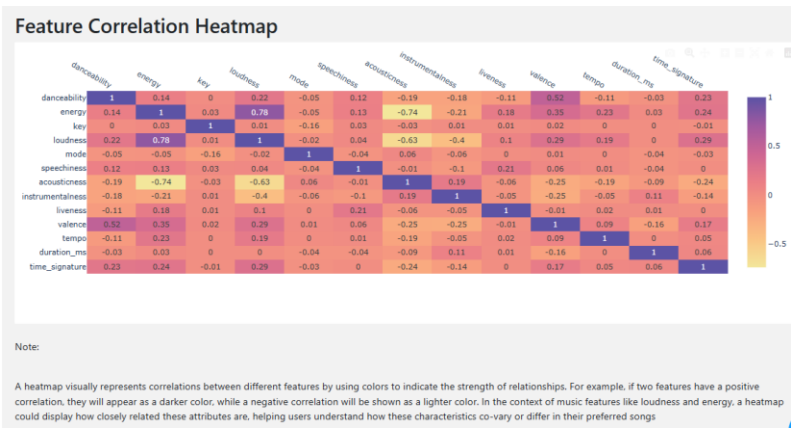


Figure 9 Feature Correlation Heatmap on dashboard

User Song Preferences Analysis: This feature employs a radar chart to illustrate and compare user preferences across various song features or characteristics within playlists or selections. Radar charts are effective for displaying multivariate data in a two-dimensional form. Users gain a holistic view of their preferences by comparing different song characteristics like danceability, energy, loudness, valence, tempo, etc., within different playlists, and identify if they have any particular preference in liking a song.

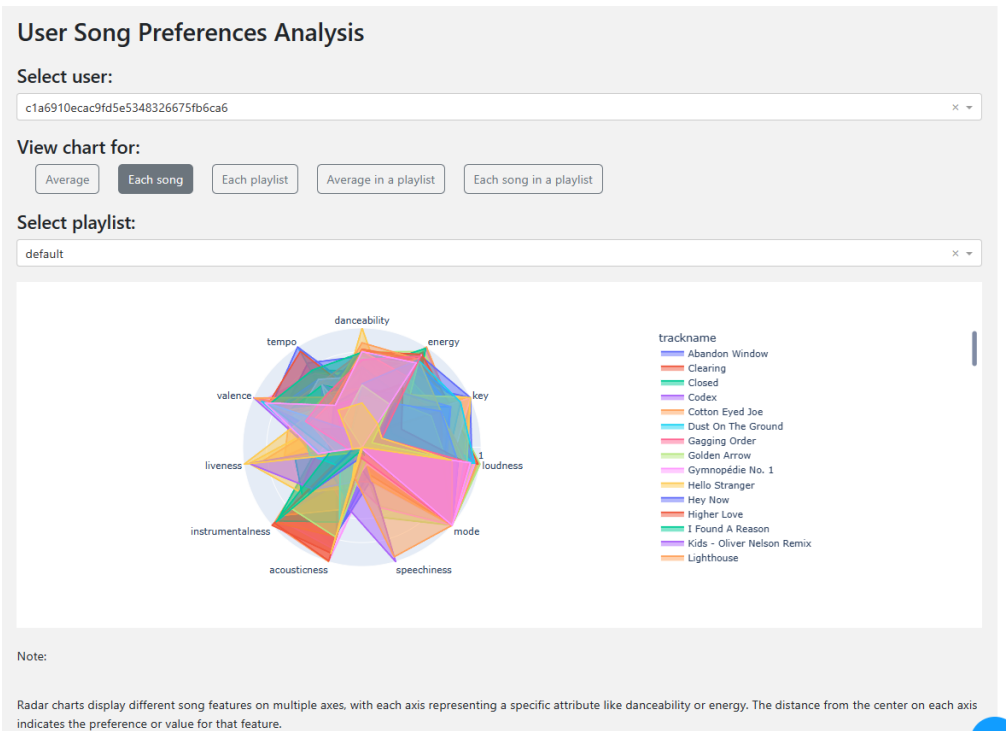


Figure 10 User Song Preference Analysis on dashboard

Similar User Matching: Two types of graphs are used for this feature. The first, a bar chart, displays precision, recall, and F1-score metrics evaluating the recommendation algorithm's performance in identifying similar users based on their music preferences. The second, a line chart, showcases evaluation metrics corresponding to different thresholds, illustrating how the algorithm's performance varies with changing thresholds. Based on the second chart, the users can choose a hyperparameter value that results in high f1-score, and precision values which would mean that the model would recommend more relevant results.



Figure 11 Similar User Matching on dashboard

Song Clustering & Exploration: For grouping songs based on selected features and identifying clusters with similar characteristics, a scatter plot is employed. Scatter plots are excellent in displaying the natural groupings of data, which can be helpful in identifying clusters or similar groups of data. This allows users to explore more songs with similar traits, aiding in identifying clusters or styles of songs.

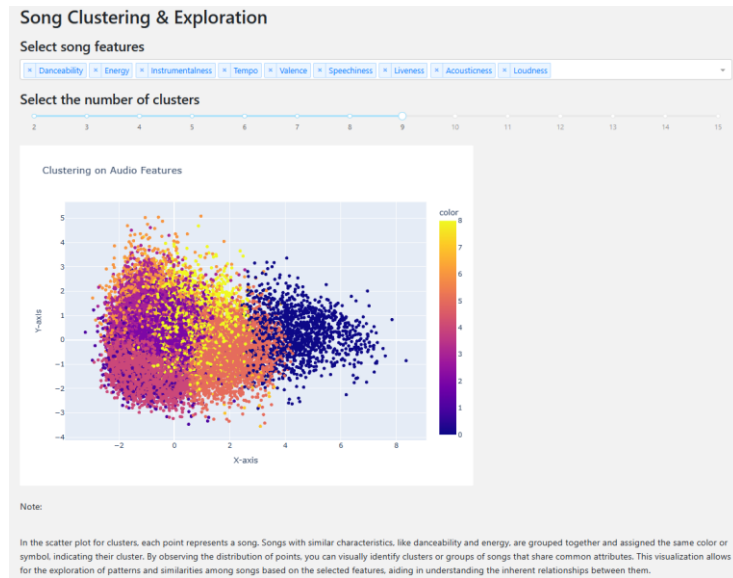


Figure 12 Song Clustering and Exploration on dashboard

Genre Identification: Several charts are used for this feature. A line chart illustrates the accuracy and loss curve, providing a visual trend of model learning. A confusion matrix offers a comparative view of predicted vs. actual genres, assessing model accuracy in genre classification. A bar chart displays predicted genres and their occurrence, helping users understand the distribution of predicted genres in the dataset. Lastly, a radar chart visualizes multiple genre preferences, depicting a user's music taste across various genres simultaneously. The first two graphs are strictly for evaluating the model's training with different learning rates where the user would want increasing accuracy curves (close to 1) and decreasing loss curves (close to 0). The last two graphs are analysis of the user's data, that is, testing the model on user's data. The bar chart shows all the genres that the user's songs fall in, whereas the radar chart shows the genre that the user is generally interested in, based on their user music profile.

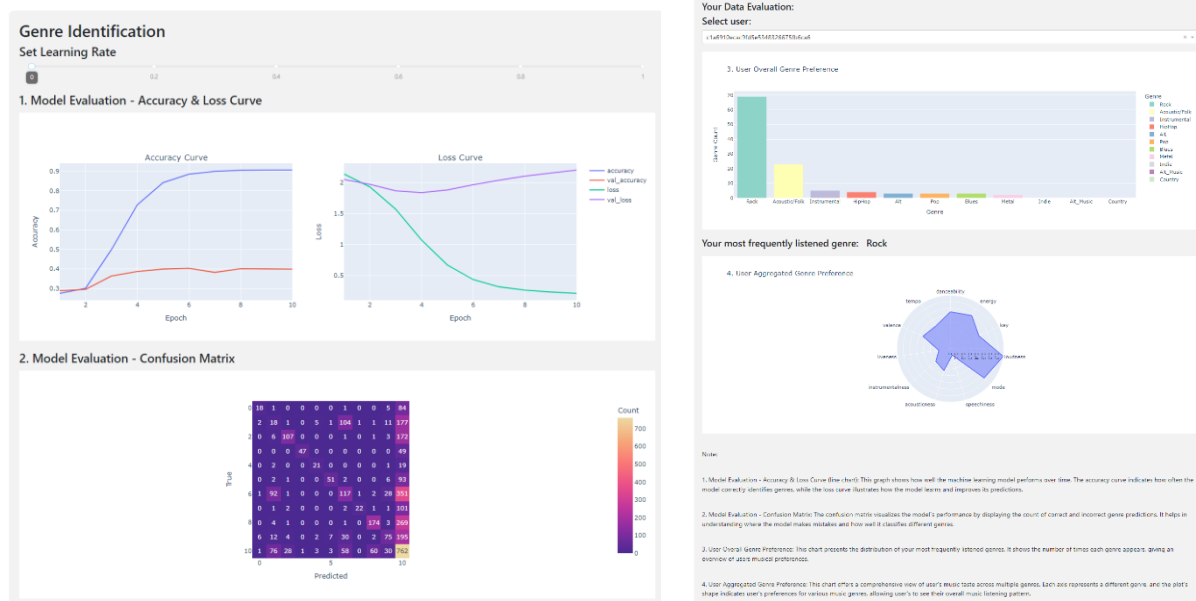


Figure 13 Genre Identification on dashboard.

5 EVALUATION

Evaluating the quality of our proposed approach involves a comprehensive analysis of the performance metrics tailored to each feature within our project. Here, we elucidate the key evaluation metrics and the methodology employed for assessing the effectiveness of our approach.

5.1 Similar User Matching

Evaluation Metrics: In assessing the performance of our recommendation system, leveraging collaborative and content-based filtering algorithms, we delve into how we define relevance, calculate key metrics, and use graphs to optimize thresholds for each algorithm.

Relevance Definition: Relevance gauges the similarity of users based on their musical preferences. For collaborative filtering, the number of common artists serves as a similarity proxy. In contrast, content-based filtering relies on song features (genre, tempo, mood), setting a threshold for the minimum number of shared features that needs to be equal or greater than the user that is being recommended to.

Metrics Utilized: Several metrics guide our evaluation process:

- **True Positives (TP):** Relevant users recommended by the system.
- **False Positives (FP):** Irrelevant users recommended by the system.
- **False Negatives (FN):** Relevant users not recommended by the system.
- **Precision:** Proportion of recommended users that are relevant. ($\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$)
- **Recall:** Proportion of relevant users that are recommended. ($\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$)
- **F1 Score:** Harmonic mean of precision and recall. ($\text{F1 Score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$)

Graphical Analysis: Graphs play a pivotal role in visualizing performance metrics across different thresholds for each algorithm. The precision, recall, and F1 score trends inform our decision on the optimal threshold. The optimal threshold strikes a balance between precision and recall, ensuring the system identifies and recommends the most relevant users while avoiding irrelevant ones.

Graph Outcomes:

1. *Collaborative Filtering:* F1 score peaks at a threshold of 5, indicating that users should share at least 5 common artists to be considered similar. This threshold optimally balances precision and recall for collaborative filtering.
2. *Content-Based Filtering:* F1 score maximizes at a threshold of 3, suggesting that users should share at least 3 common features for similarity consideration. This threshold is deemed optimal for content-based filtering.

These insights empower users to tailor the recommendation algorithm based on their preferences, ensuring a personalized and relevant user experience.

5.2 Genre Identification

Within the realm of genre identification, our evaluation metrics are tailored to measure the accuracy and performance of the neural network model. We outline how relevance is defined, detail the metrics employed, and elucidate the role of graphs in optimizing threshold selection.

Metrics Utilized: To assess the model's accuracy and classification prowess, we employ the following metrics:

- **Accuracy:** The proportion of correctly classified instances out of the total instances.
- **Confusion Matrix:** A comparative view of predicted versus actual genres, breaking down correct and incorrect classifications.
- **Genre Classification Chart:** A bar chart illustrating the predicted genres and their occurrences.

Graphical Analysis: Graphs play a pivotal role in understanding the model's performance and making informed decisions about its threshold. In this context, we emphasize the accuracy and loss curve, offering insights into model convergence and learning progress.

Graph Outcomes:

- *Accuracy & Loss Curve:* A line chart showcasing the model's learning dynamics over epochs. It provides a visual trend of the model's convergence and learning progress. Ideally, we should aim for an accuracy curve that increases and sticks close to 1, and a loss curve that decreases and sticks close to 0. These curves can help to visualize the learning process of the model, and to identify potential problems such as overfitting or underfitting. For example, if the accuracy curve shows that the accuracy increases while the validation accuracy decreases, or if the loss curve shows that the loss decreases while the validation loss increases, then it indicates that the model is overfitting. On the other hand, if the accuracy curve shows that both the accuracy and the validation accuracy are low, or if the loss curve shows that both the loss and the validation loss are high, then it indicates that the model is underfitting.
- *Confusion Matrix:* The confusion matrix is a table that shows the number of true positives, false positives, false negatives, and true negatives for each genre. A true positive is a correct prediction of a genre, a false positive is an incorrect prediction of a genre, a false negative is a missed prediction of a genre, and a true negative is a correct rejection of a genre. The confusion matrix can help to identify which genres are easy or difficult to classify, and which genres are often confused with each other. For example, if the confusion matrix shows that the model often predicts rock as metal, or vice versa, then it indicates that the model has trouble distinguishing between these two genres.

These metrics and graphical representations empower users to gauge the effectiveness of the genre identification model. It aids in understanding the model's strengths and areas for improvement, contributing to a more nuanced interpretation of the music genre predictions. Users can make informed decisions about the model's performance and adjust thresholds for personalized genre identification experiences.

5.3 Visualizations

Enhancing User Engagement: The effectiveness of our visualizations hinges on two critical aspects: interactivity and thoughtful selection to align with the problem domain. The evaluation process delves into the degree of user control and engagement facilitated by each visualization.

Interactivity Assessment: Our visualizations are meticulously crafted to offer a high level of interactivity, ensuring users have the tools to explore, analyze, and personalize their music-related insights. From scatter plots to histograms, each visualization permits user-driven exploration, allowing for a dynamic and engaging experience. Features such as hover-over details, zooming capabilities, and filter

options empower users to interact with the data, fostering a deeper connection with their music preferences.

User-Friendly Controls: The selection of visualizations is underpinned by the problem at hand, ensuring that each visualization serves a purpose in conveying relevant insights. For instance, scatter plots are employed for tempo and valence analysis to portray the correlation between song speed and emotional tone. Histograms showcase the distribution of song durations, offering clarity on engagement patterns.

Multiple Perspectives for Depth: To cater to diverse user preferences and analytical needs, we provide multiple perspectives within a single visualization. Users can choose different levels of depth, exploring various facets of their music preferences. This multifaceted approach adds depth to the user experience, allowing them to glean insights from different angles and uncover nuanced patterns within their playlists.

In essence, our visualizations are not just static representations but dynamic tools that users can wield to tailor their music exploration journey. By combining interactivity, purposeful selection, and multiple perspectives, our visualizations aim to transform data into a personalized, insightful, and visually engaging narrative for users.

6 CONCLUSION

"Harmonizing Hearts" represents a paradigm shift in online dating, transcending conventional matchmaking approaches. By decoding the language of music, we aspire to create a more harmonious and melodious digital dating space. Our project not only enriches the online dating experience but also contributes to a deeper understanding of how music can forge meaningful connections in the digital age. Through innovative visualization and machine learning techniques, we pave the way for a more personalized and emotionally resonant online dating journey. Throughout this project, several invaluable lessons were derived from the analysis conducted on music data. The project unveiled the profound emotional impact of music, recognizing its ability to evoke diverse emotions in listeners. Additionally, it showcased the wide-ranging preferences among users, revealing tendencies towards songs with similar feature values or duration. The significance of recommendation algorithms in the music domain became evident, especially with the introduction of relevance to enhance user recommendations. Furthermore, the exploration of genre identification highlighted the challenges and the need for improved datasets and model optimization for more reliable genre classification.

The analysis of various features reveals intriguing patterns within user music preferences. The analysis of the Tempo vs. Valence illustrates the diverse musical tastes within playlists, with users showcasing varying emotional preferences for different playlists, evident in the slope patterns—positive, negative, or straight lines. Song duration preferences exhibit a notable consistency, with users favoring tracks of similar lengths within playlists, reflecting a versatile musical inclination. User Song Preference Analysis reinforces the inclination towards songs with similar feature values, observable across different playlists. In the realm of Similar User Matching, a future integration of collaborative and content-based filtering promises a nuanced recommendation system, striking a balance between specificity and generalizability. The precision-recall-F1 score trends guide the selection of an optimal threshold, crucial for precision-recall equilibrium. Song clustering's emphasis on customizable features promotes transparency, fostering user interaction and exploration of similar songs. Genre identification insights pivot around accuracy and loss curves, empowering users to comprehend model convergence and refine their genre identification experience through informed threshold adjustments. These nuanced observations collectively enhance the user's understanding and engagement with the recommendation system, ensuring a tailored and insightful musical journey.

Despite the valuable insights derived, our project encountered notable limitations. The primary challenge revolved around the paucity of existing work within the domain. This prompted us to curate custom datasets, leading to a substantial loss of information. Initially, the dataset stood at a staggering 4GB, posing significant challenges in handling and processing. To manage this, we had to limit the dataset to 25,000 records to ensure it was more manageable in terms of computational resources and data handling capacity. Even with this reduction, dataset size constraints persisted, significantly impacting collaborative and content-based filtering methodologies. Additionally, the complexities of song clustering and genre identification were amplified by computational intensity, dataset skewness, and constraints related to evaluation metrics. The high computational demand posed challenges in processing vast amounts of music data efficiently. Moreover, skewed distributions within the second dataset, used for training the neural networks model in genre identification, contributed to challenges in model training and accuracy assessment. These limitations highlight the need for more scalable and optimized approaches to handle large music datasets, allowing for better analysis and improved model performance.

In considering the future trajectory of our project, several key areas have emerged for potential improvement and expansion. One crucial avenue involves enhancing the visualization capabilities to offer users a more comprehensive understanding of their music preferences. This includes extending the project to visualize user positions within song clusters which could facilitate a more intuitive exploration of music similarities, empowering users to discover songs based on similar attributes and preferences. Additionally, refining datasets by incorporating a more diverse range of music samples and optimizing machine learning models can significantly augment the accuracy of genre classification. These improvements would enable the system to categorize songs more accurately into specific genres, enriching the overall user experience. Furthermore, extending the platform into a user-centric application by providing personalized data analysis based on individual music preferences could profoundly engage users and foster stronger connections with the platform. This would also mean that our machine learning models would have access to continuous, and updated song data of each user, which would grow the shared dataspace significantly. These future enhancements aim to create a more intuitive, personalized, and compelling user experience in the realms of music analysis and online dating.

7 REFERENCES

- [1] Acharya, A. (2023). Spotify Dataset [Data set]. Available: https://www.kaggle.com/datasets/abhinav1331/spotify-dataset/data?select=spotify_data.csv.
- [2] Malgi, P. (2021). Music Genre Classification [Data set]. <https://www.kaggle.com/datasets/purumalgi/music-genre-classification?select=train.csv>
- [3] SongRecommendation-collaborativeFiltering-cluster. (2022, December 14). Kaggle.com. [Online]. Available: <https://www.kaggle.com/code/shriyutha/songrecommendation-collaborativefiltering-cluster>
- [4] Recommendation systems • ranking/scoring. (n.d.). Aman.ai. [Online]. Available: <https://aman.ai/recsys/ranking/>
- [5] Aher, P. (2023, August 9). Evaluation metrics for recommendation systems — an overview. Towards Data Science. [Online]. Available: <https://towardsdatascience.com/evaluation-metrics-for-recommendation-systems-an-overview-71290690ecba>
- [6] Reddit - dive into anything. (n.d.). Reddit.com. [Online]. Available: https://www.reddit.com/r/learnmachinelearning/comments/16notit/collaborative_filtering_jaccard_similarity_vs/
- [7] Halilovic, I. (2021, July 30). Markdown for Jupyter notebooks cheatsheet - Inge Halilovic - Medium. Medium. [Online]. Available: <https://ingeh.medium.com/markdown-for-jupyter-notebooks-cheatsheet-386c05aeebed>

- [8] Zach. (2023). How to create a distribution plot in Matplotlib. Statology. [Online]. Available: <https://www.statology.org/matplotlib-distribution-plot/>
- [9] Heatmap — seaborn 0.13.0 documentation. (n.d.). Pydata.org. [Online]. Available: <https://seaborn.pydata.org/generated/seaborn.heatmap.html>
- [10] Sharma, P. (2019, August 19). The ultimate guide to K-means clustering: Definition, methods and applications. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/>
- [11] Stewart, M. (2019, June 17). Introduction to neural networks. Towards Data Science. <https://towardsdatascience.com/simple-introduction-to-neural-networks-ac1d7c3d7a2c>
- [12] Column transformer with mixed types. (n.d.). Scikit-Learn. Retrieved December 19, 2023, from https://scikit-learn.org/stable/auto_examples/compose/plot_column_transformer_mixed_types.html
- [13] Plotly: Low-code data app development. (n.d.). Plotly.com. Retrieved December 19, 2023, from <https://plotly.com/>
- [14] Selecting the number of clusters with silhouette analysis on KMeans clustering. (n.d.). Scikit-Learn. Retrieved December 19, 2023, from https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html
- [15] Sklearn.Preprocessing.OneHotEncoder. (n.d.). Scikit-Learn. Retrieved December 19, 2023, from <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html>
- [16] Sklearn.decomposition.PCA. (n.d.). Scikit-Learn. Retrieved December 19, 2023, from <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>
- [17] TensorFlow. (n.d.). TensorFlow. Retrieved December 19, 2023, from <https://www.tensorflow.org/>
- [18] Dash bootstrap components. (n.d.). Faculty.Ai. Retrieved December 19, 2023, from <https://dash-bootstrap-components.opensource.faculty.ai/>