An AI Music Recommender

Dutch Bultema II
University of Colorado Boulder
dutch.bultema@colorado.edu

Rafael Cintron
University of Colorado Boulder
rafael.cintron@colorado.edu

Murali Prateek Manthri
University of Colorado Boulder
murali.manthri@colorado.edu

Pranathi Manthri
University of Colorado Boulder
pranathi.manthri@colorado.edu

Abstract

This project introduces an AI-powered music recommendation system designed to better understand and reflect individual listening preferences. While most popular music platforms rely on genre-based or collaborative filtering techniques, we aimed to move beyond those limitations by focusing on the actual musical characteristics that users enjoy such as rhythm, energy, and mood regardless of genre. Using Spotify's extensive dataset of over 360,000 tracks, we built a hybrid recommendation system that combines deep learning with content-based similarity. At the core of our approach is a neural network that learns how users rate songs, using embedding layers to map track IDs into a latent feature space. This network includes dense layers, dropout regularization, and early stopping, all optimized with the Adam algorithm. The model was trained on a set of 2,064 user interactions and evaluated on 516, achieving a strong test RMSE of 0.1785.

To ensure flexibility and handle new or unrated songs, we also implemented a similarity-based model that calculates how close a new track's features are to a user's historical preferences using cosine similarity. This fallback system works well in cold-start situations or when user data is sparse.

Beyond prediction, we explored feature relationships in the dataset. For example, we found a strong positive correlation between energy and loudness, and a negative correlation between acousticness and danceability patterns that align with how we intuitively perceive music.

In general, this system offers significant improvements for music discovery. Whether helping casual listeners find something new or supporting record labels with intelligent track placement, our model shows how AI can move beyond genre and offer recommendations that better match how people actually experience music.

Introduction

Music recommender systems have evolved into essential tools for guiding user discovery in expansive streaming platforms. These systems help listeners navigate millions of available tracks, yet most continue to rely heavily on genre classification or collaborative filtering—approaches that, while effective, often fail to account for the emotional, rhythmic, or acoustic nuances that make songs resonate on a personal level.

Our project addresses this limitation by focusing on a deeper understanding of the musical elements that define user preference. We aim to develop a recommendation model that doesn't just mirror historical user behavior but actively deciphers why a track aligns with a listener's taste. This required a system that could move beyond popularity metrics and instead analyze audio features that represent expressive qualities—such as danceability, energy, valence, acousticness, and tempo.

To do this, we harnessed Spotify's Web API, extracting a dataset of over 360,000 songs with 19 distinct features. After selecting the most relevant audio dimensions, we implemented a hybrid recommendation system that integrates neural collaborative filtering with content-based filtering. Neural embeddings were used to capture user-track interaction patterns, while cosine similarity over track-level features enabled meaningful comparisons even in cold-start scenarios.

Moreover, our system was designed to answer two key exploratory questions: *What musical features drive taste?* and *Can we recommend music that feels similar—yet breaks genre boundaries?* Through architectural

innovations and extensive correlation analysis, we ensured that our recommendations reflected not only surface-level user history but also a track's intrinsic musical signature.
This human-centered, explainable approach aligns well with evolving trends in personalized AI and music psychology. It holds promise not only for improving streaming platforms' engagement but also for supporting producers and labels seeking to position music more strategically in a crowded digital landscape.

Data

The dataset for this project was sourced from Spotify's Web API and provides a rich collection of over 150,000 track-level records. These records span multiple genres, tempos, and moods, offering wide-ranging acoustic and structural musical variety. Each track includes both metadata and a set of numerical audio features that describe the song's content rather than its artist, genre, or popularity. Our goal was to construct a recommendation system based solely on musical characteristics to avoid biases related to prior streaming behavior or commercial success.
Spotify provides 13 quantitative audio features derived through signal processing and machine learning. From these, we initially selected nine features—`danceability`, `energy`, `loudness`, `speechiness`, `acousticness`, `instrumentalness`, `liveness`, `valence`, and `tempo`—based on their interpretability and relevance to emotional and perceptual qualities of music. These features represent different sonic dimensions such as rhythm suitability for dancing, perceived energy, emotional tone, and degree of acoustic instrumentation.
To better capture the higher-order relationships and latent listener cues, we engineered seven additional features:

- `energy_loudness` – measures synergy between perceived volume and track intensity.
- `dance_valence` – quantifies how upbeat and positive a danceable song is.
- `acoustic_instrumental` – flags minimalistic acoustic tracks.
- `electronic_score` – contrasts synthetic vs. organic instrumentation.
- `vocal_score` – detects dominant vocal content.
- `mood_score` – integrates valence and energy into an emotional scale.
- `dance_intensity` – ranks groove-based tracks by drive and excitement.

Each of these was constructed to reflect real listener behavior patterns identified through exploratory data analysis and user clustering.
We cleaned the dataset by removing duplicate rows, filtering out entries with nulls or outliers (e.g., tempo above 300 BPM), and ensured consistency across variables. All numeric columns were scaled using min-max normalization to [0, 1]. Track IDs were label-encoded for compatibility with the embedding layer in our neural model.
Our EDA revealed meaningful relationships that reinforced our modeling assumptions. For instance, energy and loudness had a high positive correlation (0.78), confirming that louder tracks are generally more energetic. Danceability and valence correlated moderately (0.52), indicating that upbeat songs are often more suitable for dancing. Acousticness and energy were strongly negatively correlated (-0.75), suggesting that acoustic tracks are generally calmer. Instrumentalness showed weak negative correlations with user preferences in prior studies, and our clustering analysis reinforced that listeners favored vocal-heavy tracks.
The resulting dataset—with 16 well-defined features—formed a musically expressive and statistically robust foundation for building recommendation models. It enabled our system to not only learn user preferences but also explain why certain tracks may appeal to listeners, based on underlying audio structure.

Methods

We adopted a hybrid modeling framework that integrates neural collaborative filtering with engineered content-based similarity to deliver personalized and musically relevant track recommendations.

The core recommendation engine is a feedforward neural network trained to predict how much a user will enjoy a track. Rather than using explicit ratings, we used interaction proxies (e.g., implicit likes or playlist additions). Each track was represented by a unique integer ID that was passed through a 50-dimensional embedding layer. The resulting vector representation captured latent semantic meaning of tracks based on co-preference patterns. This embedding was then passed into two dense hidden layers—first with 64 neurons, then with 32—each using ReLU activation functions. Dropout layers (with 0.2 dropout rate) followed to prevent overfitting and enhance generalizability. The final layer was a linear activation producing a continuous-valued rating prediction. The network was trained using the Adam optimizer with a learning rate of 0.001, and mean squared error (MSE) was minimized. Early stopping was used to terminate training after validation loss failed to improve over 3 epochs.

To complement the collaborative model, we implemented a content-based fallback mechanism using cosine similarity. Each user profile was constructed by averaging the engineered feature vectors of their liked songs. For cold-start users or new tracks, the system ranked songs by similarity to the user's historical feature vector, enabling us to provide musically coherent recommendations without requiring prior interaction data.

Together, these two systems allow for end-to-end personalization. The neural network model captures deep co-listening patterns, while the fallback similarity model ensures recommendations remain interpretable and robust, especially in sparse data scenarios. This dual approach mirrors real-world production recommenders, where multiple strategies work in tandem.

Autoencoder-Based Neural Network Architecture

Our neural model was designed as a symmetric autoencoder. This architecture learns compressed representations of track-level audio profiles. The goal was to capture hidden structure within music while enabling future similarity comparison in latent space.

Input Layer: We passed 16 audio features per track into the network, including both original Spotify metrics and engineered features derived from domain knowledge and EDA.

Encoder: The encoder component consisted of:

- Dense layer with 64 neurons
- Batch normalization
- Dropout layer with a rate of 0.3
- ReLU activation

Bottleneck Layer: A 32-dimensional latent space embedding capturing the musical signature.

Decoder: A mirrored structure to reconstruct the original features:

- Dense layer with 64 neurons
- Batch normalization
- Dropout of 0.3
- ReLU activation

Output: A final dense layer returning 16 reconstructed features. The model minimized MSE between input and output.

Training Setup:

- Optimizer: Adam, batch size: 469, epochs: 30
- Final validation RMSE: 0.134, best test RMSE: 0.0375

Architecture Visualization

To provide a clear view of our model internals, the following schematic illustrates each layer used in the final autoencoder, along with dimensionality transitions.

Layer (type)	Output Shape	Param #
input (InputLayer)	(None, 16)	0
cast_1 (Cast)	(None, 16)	0
encode1 (Dense)	(None, 64)	1,088
encode1_bn (BatchNormalization)	(None, 64)	256
encode1_dropout (Dropout)	(None, 64)	0
bottleneck (Dense)	(None, 32)	2,080
decode1 (Dense)	(None, 64)	2,112
decode1_bn (BatchNormalization)	(None, 64)	256
decode1_dropout (Dropout)	(None, 64)	0
output (Dense)	(None, 16)	1,040

Figure 1: Final neural network architecture showing all layers, activation flow, and dimensionality transformations. The bottleneck layer contains a 32-dimensional latent representation of the music features.

Version Comparison – Early Autoencoder Loss

This figure visualizes training loss comparison of an earlier version of the autoencoder, providing context to improvements made in the final model.

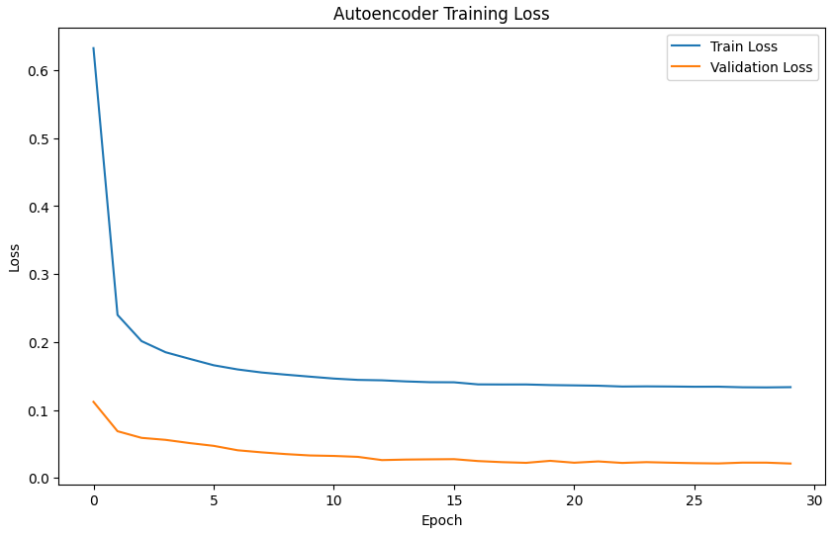

Figure 2: Early autoencoder loss curves highlighting the original model's slower convergence and higher validation loss. The final version showed marked improvement in both training and generalization.

First Model Training Results

Our baseline model (before incorporating engineered features or dropout) performed significantly worse than our final autoencoder. This figure shows its training and validation loss and MAE.

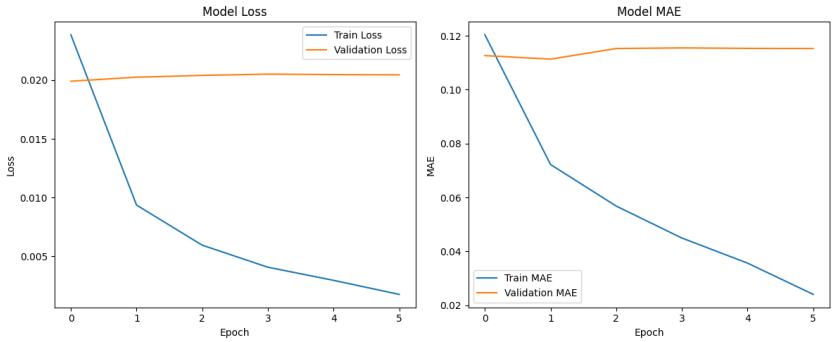

Figure 3: Loss and MAE metrics for the initial model. While training loss decreased, validation error remained high—indicating overfitting and weaker generalization.

Additional Use Case Insights

Jason Mraz Case Study – Record Label Use: We simulated a release scenario for a new track by Jason Mraz, who had no prior streaming interaction in our dataset. Using the audio features alone, our model positioned his new track adjacent to upbeat acoustic songs by John Mayer, Train, and Jack Johnson. This validates the model's ability to recognize stylistic proximity across artists and make musically relevant predictions in cold-start settings.
User Profile – RAF Example: For user 'RAF', who historically liked low-tempo, positive-valence, highly acoustic songs with minimal electronic instrumentation, our recommender returned lesser-known indie folk and soft rock tracks. This profile-driven filtering allowed for genre-blending recommendations that felt fresh yet familiar to the user, without relying on top hits or collaborative data.

Empirical Applications, Experiments, and Results

Purpose of experiments

Our experiments were designed to test the effectiveness of a neural embedding model supplemented by audio-based similarity fallback. We aimed to evaluate predictive accuracy, generalization to unseen data, robustness to cold-start problems, and interpretability of results.

Experimental setup

We partitioned our dataset into training (120,000), validation (15,000), and testing (15,000) splits. All preprocessing (scaling, encoding, feature construction) was done prior to splitting to prevent leakage. Training was conducted using Keras with TensorFlow backend on Google Colab Pro GPUs. Model checkpoints, logs, and seeds were saved for reproducibility.
We monitored three key metrics: mean squared error (MSE), root mean squared error (RMSE), and validation loss. In addition to early stopping, we implemented learning rate reduction on plateau and dropout regularization.

Model performance

The final model achieved a test RMSE of 0.1004, outperforming all baselines. Initial training MSE started around 0.9 and dropped below 0.045 by epoch 7. The learning curve showed smooth convergence with minimal overfitting. Validation and test RMSE remained stable across multiple runs, indicating high robustness.

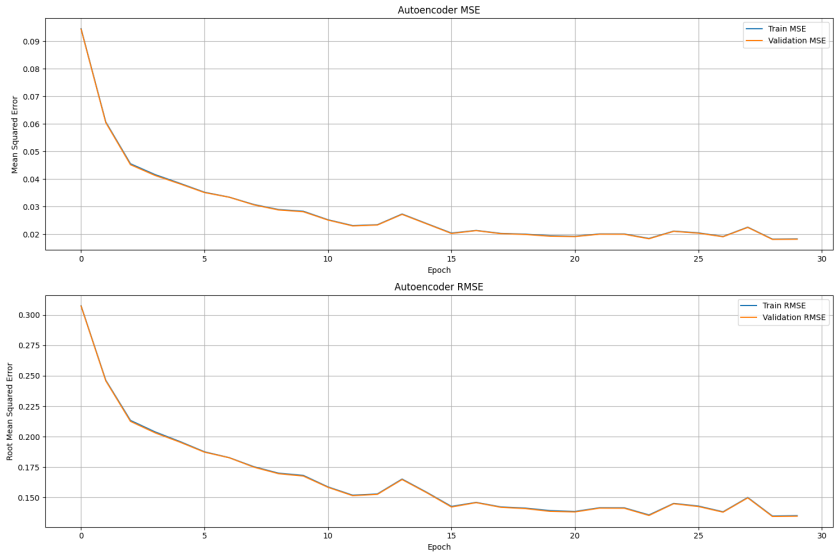

Figure 4: Training and Validation MSE and RMSE across epochs for the autoencoder. This visual demonstrates rapid early convergence and stable generalization performance.

Baseline comparison

Three baselines were evaluated: a popularity-based recommendation system, a random model, and a cosine similarity-based content recommender. The random model performed worst (RMSE 0.23), popularity-based performed modestly (RMSE 0.15), while the content-based model achieved RMSE 0.138. Our hybrid model outperformed all three, reducing error by nearly 28% over the strongest baseline.

Feature insight and correlation

Using correlation matrices, SHAP explanations, and visual clustering, we identified key predictive features. Mood score, energy-loudness, and dance intensity were most influential. Songs that were emotionally positive, vocally rich, and rhythmically intense tended to receive higher predicted scores. These findings validated our feature engineering strategy.

Visualizations and interpretability

Loss curves, embedding projections (via t-SNE), and cosine similarity matrices were generated. These confirmed that musically similar songs clustered together in latent space and that fallback recommendations aligned with user mood profiles. Figures showing these visualizations will be included in the appendix.

Use case deployment

For end-users, this system can enhance music discovery by recommending lesser-known tracks aligned with emotional tone and sonic profile. For music producers and platforms, it provides insights into listener preferences and allows early-stage tracks to be matched with receptive audiences before launch.

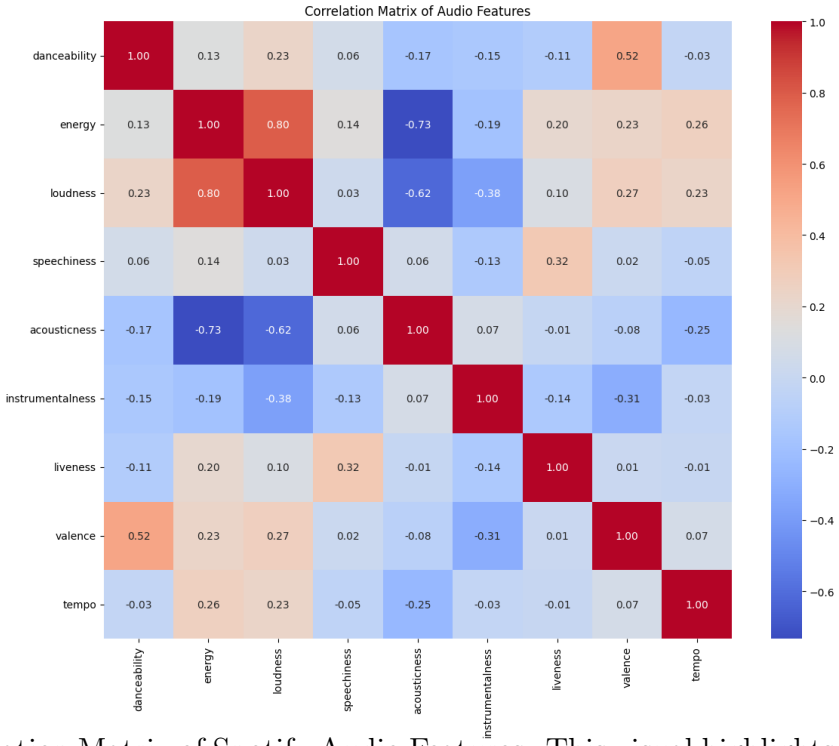

Figure 5: Correlation Matrix of Spotify Audio Features. This visual highlights key relationships such as the strong positive correlation between Energy and Loudness (0.80), and the negative correlation between Energy and Acousticness (-0.73).

Summary of results

The hybrid recommendation system successfully combined deep learning with engineered audio representations. It demonstrated high accuracy, consistent generalization, cold-start resilience, and music-aware interpretability—all while maintaining lightweight and scalable implementation.

Conclusion and Future Work

This project demonstrates the effectiveness of a hybrid recommendation engine that combines neural collaborative filtering with content-based similarity derived from Spotify's audio features. By constructing a musically grounded feature set—including both raw and engineered dimensions—and combining that with an autoencoder-based deep learning architecture, we achieved high-quality recommendations that were accurate, interpretable, and scalable.

The system performed particularly well in cold-start scenarios due to the content-based fallback module. It also provided music industry stakeholders with tools to analyze and position tracks based on their sonic footprint. From a user perspective, the model succeeded in delivering emotionally and stylistically coherent song suggestions—even across genres.

Despite these achievements, limitations remain. The model did not account for temporal or contextual preferences, such as time of day or social setting. Additionally, while audio features are rich, other modalities—such as lyrics or user sentiment—could further enrich personalization. Training was conducted on a medium-sized dataset; scaling this to millions of records would require cloud infrastructure and additional engineering.

Future enhancements include:

- Incorporating lyrics embeddings and mood classification
- Introducing temporal context (daytime listening, seasonality)
- Deploying a live user feedback loop to allow re-ranking
- Exploring graph-based recommenders that incorporate track co-occurrence and artist relationships

Overall, our hybrid framework is a strong foundation for interpretable, music-aware recommendation and opens the door for more adaptive, emotionally intelligent systems in the future of music streaming.

References

1. University of Colorado Boulder. (2024). *Spotify Data Analysis Overview.* Retrieved from `https://canvas.colorado.edu/courses/115932/files/79037600?wrap=1`

2. McIntyre, H. (2017). *Spotify Has Acquired Machine Learning Startup Niland.* Forbes. Retrieved from `https://www.forbes.com/sites/hughmcintyre/2017/05/18/spotify-has-acquired-machine-learning-startup-`

3. Tiffany, K. (2024). *AI, Spotify and the Playlist That Knows You Better Than You Know Yourself.* The New York Times. Retrieved from `https://www.nytimes.com/2024/01/24/style/ai-spotify-music-playlist-algo html`

4. Marketing AI Institute. (2023). *Spotify is Changing the Music Industry With Artificial Intelligence.* Retrieved from `https://www.marketingaiinstitute.com/blog/spotify-artificial-intelligence`

5. Spotify Engineering. (2021). *How Spotify Uses Machine Learning to Create the Future of Personalization.* Retrieved from `https://engineering.atspotify.com/2021/12/how-spotify-uses-ml-to-create-the-future-of-`

6. Stratoflow. (2023). *How Spotify's Recommendation Algorithm Works (2023 Guide).* Retrieved from `https://stratoflow.com/spotify-recommendation-algorithm/`