IS 733 Data Mining
Project proposal
Group 3

Title: Personalized Song Recommendation System using Spotify Audio Features

Team Members:

Anirudh Kakumani
Ankit Kumar Nath
FNU Priyanka Rani
FNU Trupti Gangji
Zeeshan Mohammed

Abstract

In an era where music streaming platforms host millions of tracks, users often face difficulty in discovering songs that match their personal taste. This project proposes a personalized music recommendation system based on a content-based filtering approach using Spotify audio features. The system analyzes key audio attributes such as valence, energy, danceability, and tempo to understand user preferences and recommend similar songs.

As a starting point, we use the valence attribute to classify songs into "liked" and "not liked," and apply machine learning models—K-Nearest Neighbors (KNN) and Random Forest—to learn patterns in audio features that influence song likability. Evaluation metrics such as precision, recall, and F1-score are used to assess model performance.

The proposed system also lays the foundation for dynamic recommendation capabilities. In future iterations, users will be able to select genres they prefer, and the system will recommend not only similar songs but also newly added tracks that align with the user's taste profile. Users will also be able to get recommendations based on their liked songs. By incorporating additional target variables like track_genre and popularity, the model aims to deliver more diverse, personalized, and explainable music recommendations. Built entirely using Google Colab, the system is designed to be scalable, interactive, and ready for real-world application.

Business Understanding / Problem Statement

With millions of songs available on platforms like Spotify, users often struggle to discover music that fits their personal taste. Most existing recommendation systems rely on user history or popularity, which doesn’t always help new users or songs get noticed.

Our project aims to solve this by building a recommendation system that uses the actual audio features of songs to find and suggest similar tracks. This content-based approach makes it possible to recommend songs even for new users or newly added music, making the experience more personalized and helpful from the start.

1. Main Objectives

- To build a music recommendation system that suggests songs based on a user’s audio preferences instead of relying on user ratings or streaming history.

- To use a content-based approach that focuses on the characteristics of songs, like valence, energy, and danceability, to identify and recommend similar tracks.

- To experiment with machine learning models (like KNN and Random Forest) to classify songs as liked or not liked, based on their audio features.

- To explore how we can later improve recommendations by including genres or popularity as additional targets.

- To make the system capable of handling new songs by recommending them based on their similarity to a user's taste and selected genre.

2. Dataset Details

Source: Spotify Tracks Dataset
(https://huggingface.co/datasets/maharshipandya/spotify-tracks-dataset)

Total Records: 114,000 rows
Total Attributes (Columns): 21
Selected Features for Modeling: 9 key audio features

- danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, tempo, duration_ms

Target Variable(s)

1. Current Target: Valence-Based Binary Label
We used valence (which reflects how positive or happy a song feels) to divide the data into two classes:

- Liked → valence > 0.5
- Not Liked → valence ≤ 0.5

This binary label helps us start with a basic model that can predict song likability using audio features.

2. Future Targets (Planned)

a. track_genre:
- Will help us train models that recommend songs based on genre.
- Can be used to filter songs by genre preferences like Pop, Jazz, Lo-fi, etc.

b. Popularity:
- A Spotify-provided numeric score based on how often a song is played.
- Can be useful for recommending trending or widely-liked songs.

- We’re considering turning this into labeled categories like "Low", "Medium", and "High" popularity.

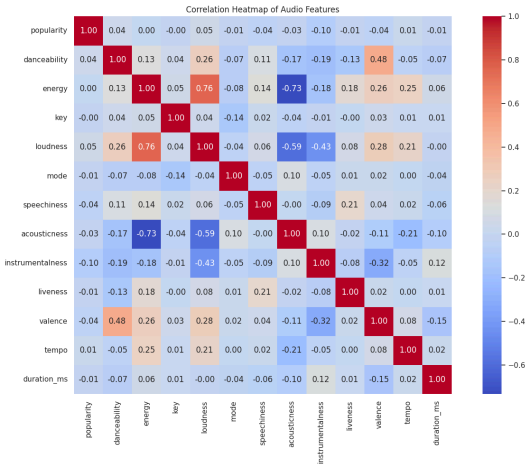
3. Exploratory Data Analysis (EDA)

- Summary Statistics:

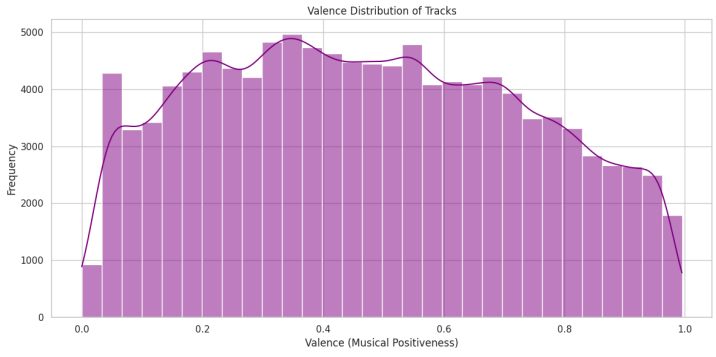
Features	Mean	Std Dev	Min	Max
Danceability	0.57	0.17	0.00	1.00
Energy	0.64	0.25	0.00	1.00
Loudness (dB)	-8.25	5.03	-49	3
Acousticness	0.31	0.33	0.00	1.00
Instrumentalness	0.15	0.30	0.00	1.00
Valence	0.47	0.26	0.00	1.00
Tempo (BPM)	122.14	29.98	0.00	244

eability.

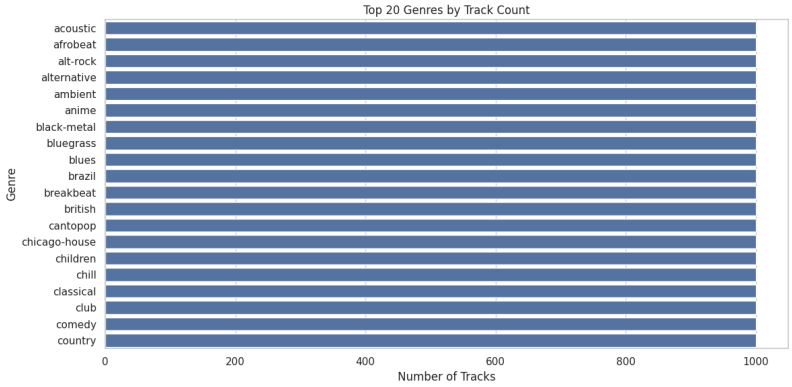
- Visualizations Included:
    - Correlation Heatmap (to identify multicollinearity):



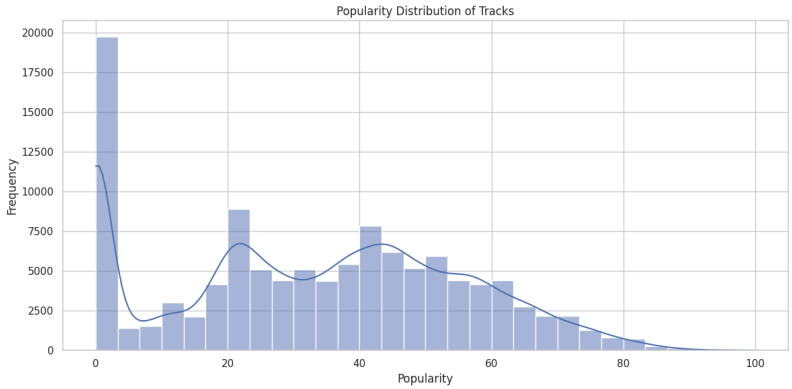
    - Valence Distribution:



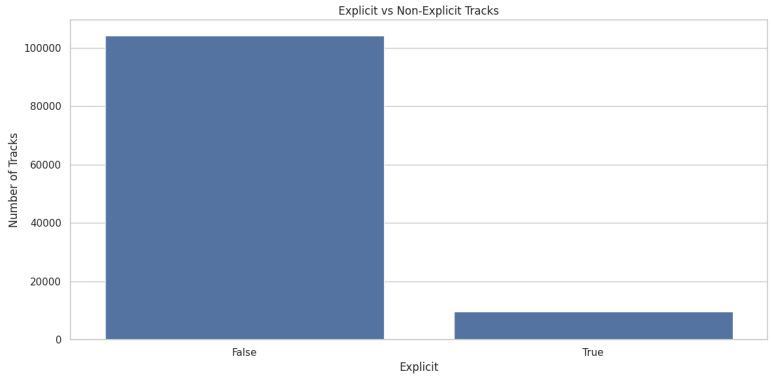
- Genre Distribution:



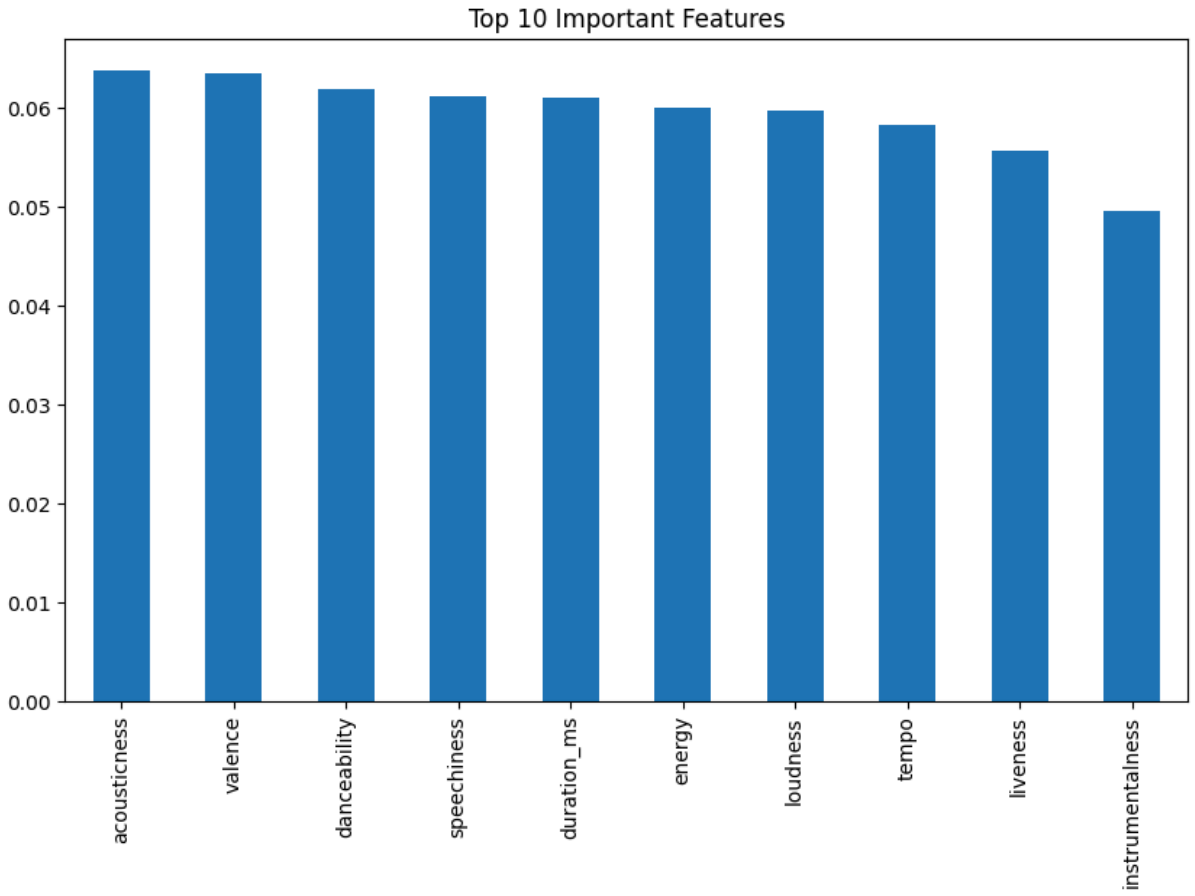
- Popularity Distribution:



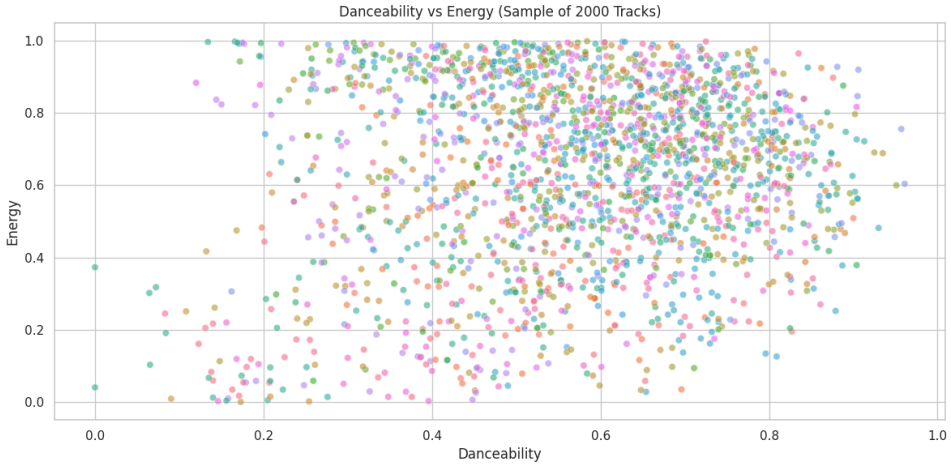
- Explicit vs Non-Explicit



- Top 10 features:



- Danceability Vs Energy



Data Preprocessing: Now, we performed following steps for preprocessing

- Dropped missing values.
- Created binary labels from the continuous valence feature.
- Selected 9 relevant features for training.
- Applied standardization using StandardScaler to prepare data for KNN.
- Split the dataset into training and testing sets (80/20 split).

In future iterations,, we plan to clean and standardize genre labels for better genre-based filtering. We also aim to categorize the popularity feature into levels (e.g., low, medium, high) to simplify future classification tasks. If we introduce popularity or multi-class labels, binning and feature transformations may also be used to improve model performance and interpretability. For visualization or clustering tasks, dimensionality reduction techniques like PCA may also be explored.

4. Data Mining Techniques & Why

Content-Based Filtering Approach: For this project, we decided to go with a content-based recommendation method. The idea is to recommend songs that are similar in terms of their audio features (like tempo, energy, and valence) to songs the user already likes.

Why this approach?

- Doesn't need user history or ratings, which makes it easier to work with.

- Works even when users or songs are new (cold start).
- Makes use of the actual audio characteristics provided by Spotify.
- It's explainable — we know *why* a song is being recommended based on feature similarity.

We now used for our initial understanding 2 models: KNN and Random forest classifier.

1. K-Nearest Neighbors (KNN)

   We used KNN both for classifying whether a song would be liked or not, and for finding similar songs.
   - Used StandardScaler to scale features
   - Accuracy: 96.76%

   Why KNN?

- It fits well with our content-based idea — it checks how similar a song is to others.
- It's simple and doesn't require training in the traditional sense.
- Easy to understand and implement.
- It works really well when the features are numerical and scaled properly.
- Evaluation Metrics Used: Accuracy, Confusion Matrix, Precision, Recall, F1-Score

2. Random Forest Classifier

   We also tested Random Forest for binary classification (liked vs not liked).
   - No scaling needed for this model
   - Accuracy: 100%

   Why Random Forest?
   - Can handle complex relationships between features.
   - Gives us feature importance which is helpful to understand what matters most.
   - It performed really well — though the 100% accuracy probably means it's overfitting.

   Note: That perfect accuracy is usually overfitting, so we're planning to use cross-validation and more robust testing to check if the model is genuinely performing well or just memorizing the training data.

5. Evaluation Metrics

To evaluate how well our models performed, we used several common classification metrics:

- Accuracy – Measures how many total predictions were correct out of all predictions.
- Precision – Of the songs predicted as "liked," how many were actually liked.
- Recall – Of all the songs that were actually liked, how many the model was able to correctly identify.
- F1 Score – Combines precision and recall into one score that balances both.

We also used a confusion matrix to break down the predictions and see exactly where the model got things right or wrong. It helped us understand:

- False Positives – Songs that were predicted as liked but weren't.
- False Negatives – Songs that were actually liked but predicted as not liked.

These metrics together gave us a more complete picture of the model's performance — not just how often it was right, but how well it handled both correct and incorrect predictions.

6. Next Steps / Future Work

We've got a solid base working with content-based filtering and classification, but there's still a lot we want to improve and explore as we move forward. Here's what we're planning next:

- **Improve Content-Based Recommendations:** We plan to compute cosine similarity between songs to find the top-N tracks that are most similar to a user's favorites.
  We also want to build user profile vectors by averaging the features of songs the user likes. This way, we can recommend songs that closely match their overall taste.
- **Add Genre and Mood Filters:** We want to give users more control by letting them pick a genre or mood they're in the mood for.
  By using features like track_genre, valence, and energy, we can tag songs accordingly (e.g., relaxing, upbeat, energetic) and make more personalized suggestions based on those filters.
- **Try Additional Models:** Right now, we've used KNN and Random Forest, but we're planning to try out a few other models like Logistic Regression, SVM, or XGBoost to see if we can improve performance or gain new insights.
  We're also considering PCA or similar techniques to reduce dimensions and help visualize how songs cluster based on their features.

Conclusion

This project sets the foundation for a personalized music recommendation system that focuses on what really matters — the sound and feel of the music itself. By using audio features and a content-based approach, we aim to help users discover songs that match their taste without relying on ratings or popularity. As we continue to improve the model by adding genre filters, user profiling, and new features, the goal is to create a recommendation system that feels more personal, intuitive, and adaptable to individual preferences.