Spotify Recommender System

By Vibe Verse

Authored by:
1. Yusuf Ali
2. Alvin Asingo
3. Vallary Banda
4. Michael Achoki

Team Members

Yusuf Ali(Modeling)

Vallary Banda(pre-processing and EDA)

Alvin Asingo (Documentation)

Michael Achoki(Presentation)

Outline

1. Overview
2. Business Understanding
3. Data Understanding
4. Exploratory Data Analysis
5. Modelling
6. Evaluation
7. Challenges Encountered
8. Conclusions
9. Recommendations

Overview

- ❖ Project Objective - to develop an effective music recommendation system that enhances user satisfaction by providing personalized track suggestions aligned with individual listening preferences.
- ❖ Dataset - Over 114,000 tracks , audio features and genre classification from Spotify
- ❖ Sentiment Labels -

Business Understanding

Music consumption has changed significantly in today's digital world, with consumers increasingly depending on tailored suggestions to find new musicians and songs. Since the rise of streaming services has changed how people listen to music, platforms must be able to properly recommend songs in order to stand out from the competition. A well-thought-out music recommendation system may greatly improve user experience by boosting engagement and encouraging brand loyalty. Our methodology seeks to offer personalized recommendations that appeal to individual interests by examining user preferences, listening patterns, and contextual information. In addition to satisfying customers' varied musical tastes, this increases retention rates, which eventually helps the platform succeed in a cutthroat industry.

Key Objectives:

- Identify optimal models.
- Deploy a web application
- Enhance User Engagement and Retention.

Data Understanding.

The dataset used in this project contains detailed information on 114,000 tracks and includes various audio features, track metadata, and genre classifications. The primary goal is to leverage these features to create a recommendation system that can suggest similar tracks based on user preferences.

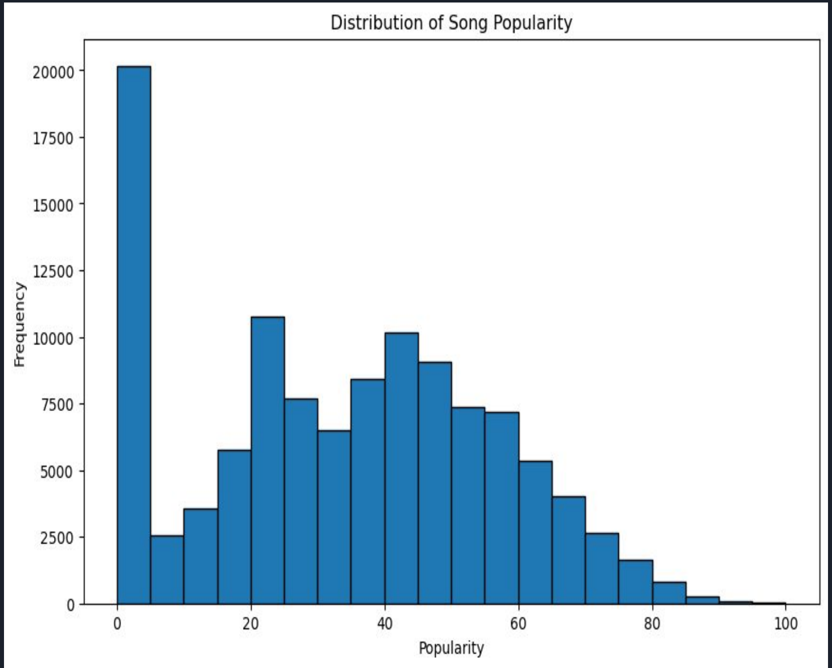
Track Metadata:

- track_id: Unique identifier for each track.
- artists: Names of the artist(s) associated with each track.
- album_name: Name of the album in which the track appears.
- track_name: Title of the track.
- track_genre: Genre classification of each track, providing valuable context for similarity recommendations.

Audio Features:

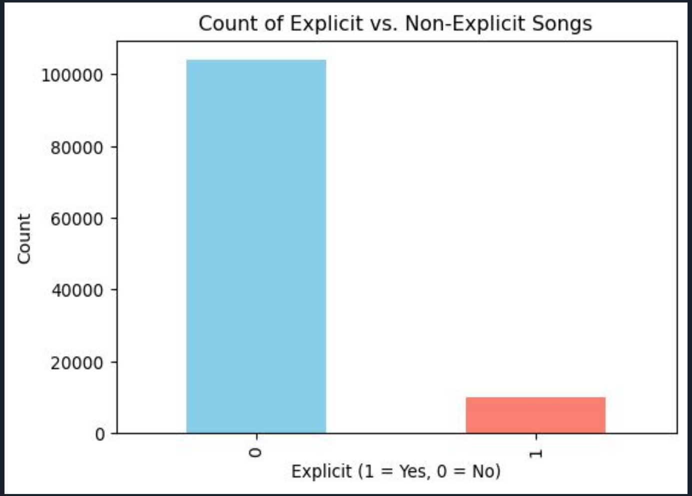
- popularity: Popularity score (integer), which can indicate user preference or track familiarity.
- duration_ms: Duration of the track in milliseconds, a basic feature that can affect user preference.
- liveness: Float value indicating the presence of a live audience.
- valence: Float value that captures the positivity of a track, often linked to emotional tone.
- tempo: Beats per minute (BPM) of the track, indicating its overall pace.

Exploratory Data Analysis



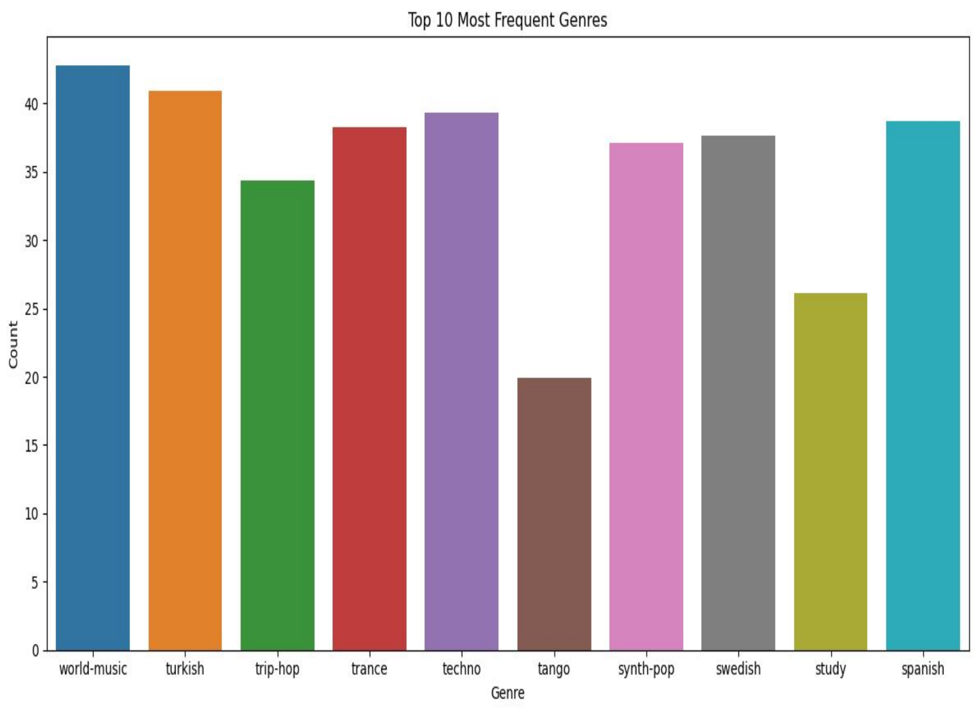
The plot reveals a high frequency of songs with low popularity (near zero), suggesting that a large portion of the dataset consists of less popular tracks. As popularity increases, the frequency of songs decreases, indicating that only a small subset of songs achieves high popularity scores.

Count of Explicit vs. Non-Explicit Songs.



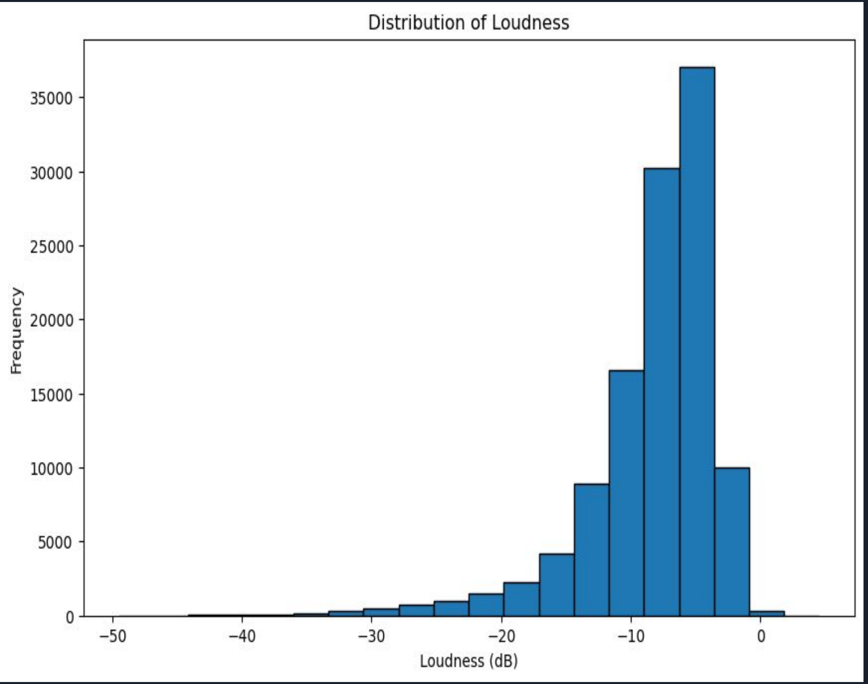
The vast majority of songs are non-explicit, with only a small proportion marked as explicit. This distribution may reflect a preference for cleaner content or limited explicit content across most songs in the dataset.

Top 10 Most Frequent Genres



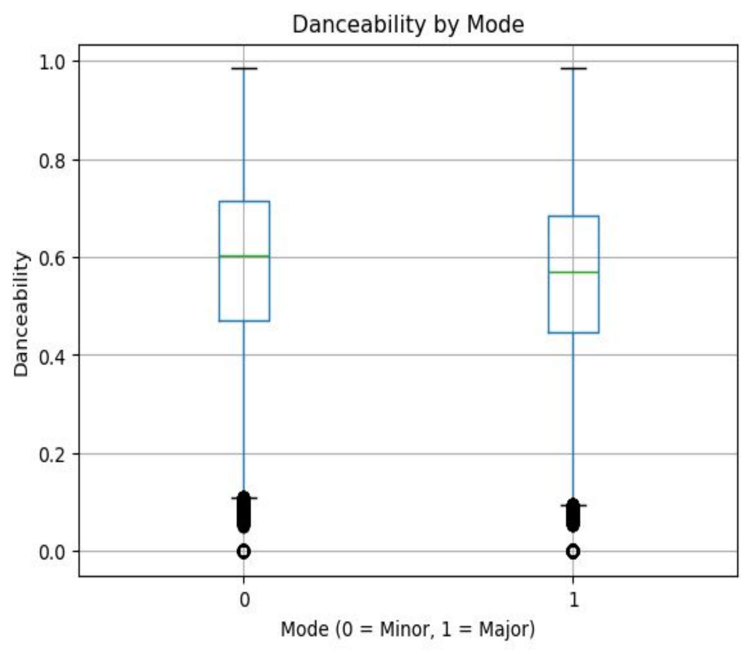
The approximately mean frequency of each genre suggests that these popular genres are adequately represented. This equilibrium enables the recommendation algorithm to provide a variety of suggestions for various musical genres.

Distribution of Loudness



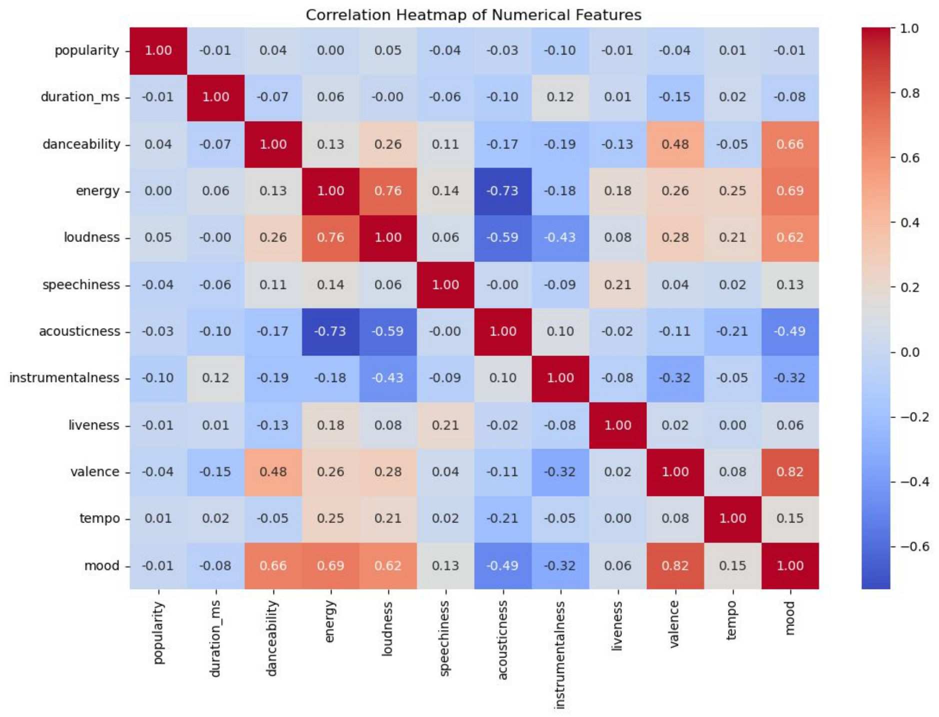
With a tendency toward softer music (lower loudness values), most songs have loudness ratings between -20 dB and -5 dB. Extremely loud songs are rare; most are either very mild (below -30 dB) or almost 0 dB.

Danceability by Mode



The plot shows the distribution of danceability scores for songs in different modes, where 0 represents a minor key, and 1 represents a major key. The median danceability is similar across both modes, with no significant difference between minor and major modes. The spread and range of danceability scores are also comparable, indicating that a song's mode (major or minor) does not have a strong influence on its danceability.

Correlation Heatmap of Numerical Features



Loudness and Energy: Energy and loudness have a high positive association (0.76), indicating that more energetic songs are typically louder. Since mood is partially generated from valence, it makes sense that there is a strong connection (0.82) between valence and mood. Songs with a higher mood score are typically more upbeat. Acousticity and Loudness: Acousticity is negatively correlated with both energy (-0.73) and loudness (-0.59), suggesting that songs that are more acoustic tend to be quieter and less vivacious.

Modelling

Cosine Similarity-Based Recommendation Model

- Goal: Recommends songs that align in genre and musical qualities, creating a cohesive listening experience.
- How It Works:
    a. Song Analysis: Combines each song's genre and musical characteristics (like rhythm and energy) into a unique profile.
    b. Matching Process: Finds songs with closely matching profiles, focusing on those with a similar feel to the input song.
    c. Personalized Suggestions: For each input song, the model ranks the closest matches to provide a relevant list of recommendations.

Results:

- Precision: 83% – Most recommendations align with the genre of the input song, ensuring high relevance.
- Ranking Quality (MRR): 0.90 – Relevant tracks appear consistently near the top, helping listeners find the best matches quickly.

Modelling contd’

Model 2: Autoencoder-Based Track Recommendation

- **Goal**: Recommends songs by identifying underlying patterns in musical characteristics, allowing for more personalized suggestions.
- **How It Works**:
  - **Feature Compression**: Uses an autoencoder to create a compact “profile” of each song by capturing its core characteristics.
  - **Matching Process**: Finds similar songs by comparing these profiles, ensuring recommendations that share a similar essence.
  - **Personalized Suggestions**: Provides recommendations that align well with each input song’s unique features.

Results:

- **Training Loss**: 0.0009 – Shows that the model accurately learned and captured important song features during training.
- **Validation Loss**: 0.0012 – Indicates that the model generalizes well to new songs, retaining accuracy without overfitting.
- **Average Similarity**: 0.9969 – Reflects a very high similarity between each input song and its recommended tracks, confirming that the recommendations closely match the style and feel of the input.

Modelling contd’

Model 2: Autoencoder-Based Track Recommendation

- **Goal**: Recommends songs by identifying underlying patterns in musical characteristics, allowing for more personalized suggestions.
- **How It Works**:
  - **Feature Compression**: Uses an autoencoder to create a compact “profile” of each song by capturing its core characteristics.
  - **Matching Process**: Finds similar songs by comparing these profiles, ensuring recommendations that share a similar essence.
  - **Personalized Suggestions**: Provides recommendations that align well with each input song’s unique features.

Results:

- **Training Loss**: 0.0009 – Shows that the model accurately learned and captured important song features during training.
- **Validation Loss**: 0.0012 – Indicates that the model generalizes well to new songs, retaining accuracy without overfitting.
- **Average Similarity**: 0.9969 – Reflects a very high similarity between each input song and its recommended tracks, confirming that the recommendations closely match the style and feel of the input.

Modelling contd’

Model 3: Transformer-Based Track Recommendation

- **Goal**: Recommends songs by capturing complex relationships between musical features, creating relevant and genre-aligned suggestions.
- **How It Works**:
  - **Feature Processing**: Prepares each song’s genre and musical traits as inputs, combining these into a feature matrix for analysis.
  - **Embedding Creation**: Uses a Transformer model to generate a compact profile for each track, focusing on core relationships.
  - **Finding Similar Songs**: Compares these profiles to find tracks with matching characteristics, ensuring genre consistency.

Results:

- **Precision**: 0.76 – Most recommendations align with the input track’s genre, providing a cohesive listening experience.
- **MRR**: 0.90 – Ensures relevant tracks appear near the top of each recommendation list, allowing users to find the best matches quickly.

Evaluation

1. Deep Learning Model

- **How It Works**: This model learns the core characteristics of each song to understand what makes them unique.
- **Performance**:
  - **High Accuracy**: The model excels at finding songs that are very similar to your favorite tracks.
  - **User Relevance**: Recommendations closely match the input songs, providing a highly personalized experience.

2. Transformer-Based Model

- **How It Works**: This model captures complex relationships between different song features, including genre and musical elements.
- **Performance**:
  - **Good Accuracy**: It offers relevant song suggestions, though with slightly less genre alignment compared to the first model.
  - **Variety in Recommendations**: Introduces a mix of songs, including some that might be outside your usual genre, helping you discover new music.

Deployment

The recommendation system was deployed on Streamlit, providing an interactive interface where users can input up to 5 songs and specify the number of recommendations for each. The backend is powered by Flask, which efficiently handles requests and processes the cosine similarity model to generate personalized music recommendations in real-time. This setup offers users a seamless experience with flexibility in choosing the number of recommendations for each track.

Challenges Encountered

1. **Data Sample Size**:
   - Due to memory limits, we used smaller samples from the original dataset, which may reduce the variety and diversity of recommendations.
2. **Model Differences**:
   - The two models we used have different ways of matching songs. This required separate evaluation methods, making it challenging to compare them directly.
3. **Training Limits**:
   - The deep learning model had limited training time to prevent overfitting. With more resources, longer training could improve its ability to capture complex song relationships.
4. **Genre Imbalance**:
   - Some genres are underrepresented in the dataset, which could lead to biased recommendations favoring more common genres.

Conclusions

This project successfully demonstrates how a music recommendation system can provide relevant and personalized song suggestions by analyzing both musical features and genre. Despite some limitations, the system achieves its core goal: delivering recommendations that align well with the listener's preferences.

Key Takeaways:

- **Effective Personalization**: The models used offer accurate, genre-aligned recommendations, enhancing the user's listening experience.
- **Scalability**: The project lays a foundation for scaling recommendations to larger datasets, highlighting how similar systems could work on a broader scale.
- **Future Improvements**: Addressing data size and genre diversity could further enhance recommendation quality and fairness across all music styles.

Overall, this project highlights the potential for advanced recommendation models to provide tailored and engaging music experiences.

Recommendations

1. **Personalized Recommendations**
    - Use tailored playlists and genre-focused suggestions to engage various user segments, such as new listeners or genre explorers. Personalized recommendations can encourage users to return and discover more.
2. **Utilize User Interaction Data**
    - Integrate feedback on actions like skips, favorites, or time spent on tracks to refine recommendations. This adjustment helps align suggestions with user preferences, making them more relevant.
3. **Adapt to Trends**
    - Regularly adjust recommendations to reflect popular genres or moods, keeping the platform current and aligned with listener interests. This can increase user satisfaction and keep the platform feeling fresh.
4. **Retrain Models Regularly**
    - Retrain the recommendation models periodically using recent data to capture evolving music tastes. This keeps recommendations relevant and competitive in a dynamic market.
5. **Create a Feedback Loop**
    - Enable users to provide feedback on recommendations, such as track ratings or playlist relevance. Continuous feedback allows for fine-tuning and improvement, making the experience more user-centered.

Summary

This project developed a music recommendation system designed to provide personalized and genre-aligned song suggestions. Using three models—a cosine similarity model, an autoencoder model, and a Transformer model—the system identifies songs that closely match a user's selected track based on musical features and genre. Each model offers a unique approach, from focusing on genre accuracy to balancing relevance and diversity, ensuring that recommendations meet varied user preferences.

Evaluation metrics, including precision, ranking accuracy (MRR), and similarity, confirm that the models deliver relevant and engaging song suggestions. While effective, future improvements could incorporate user interaction data to further personalize recommendations and adapt to emerging music trends, creating a dynamic experience that strengthens user engagement and loyalty.

Thank You !!!!