Music Recommendation System

Final Project Presentation

ESHWAR VANGURI - 23B0916
PUNIT KUMAR - 23B0996

4 May, 2025

Problem Statement

Input:

- User's favorite songs
- Optional weights for each song in the history (for personalized recommendations)

Output:

- List of recommended songs based on audio features and similarity metrics
- Optional cluster-based recommendations

Example:

- Input: "Shape of You", "Blinding Lights", "Lovers Rock"
- Output: Similar songs with artist names, similarity scores, and genres

The project aims to build a recommendation system that suggests songs based on a user's listening history using machine learning techniques.

Motivation

Personalization:

Music streaming platforms need effective recommendation algorithms to enhance user experience

Discovery:

Help users discover new music aligned with their preferences

Application:

Practical application of clustering and similarity algorithms to a real-world problem

Challenge:

Working with high-dimensional audio feature data and creating meaningful recommendations

Literature Review

Content-Based Filtering:

Our approach builds on content-based filtering techniques that use item features

K-means Clustering:

Inspired by research on using clustering for content discovery

Cosine Similarity:

Widely used in recommendation systems for measuring content similarity

Key papers: "Music Recommendation Using Content-Based and Collaborative Filtering Methods" (Lee et al.), "An Efficient Approach for Content-Based Recommendation Systems Using K-means Clustering" (Takata & Chiyonobu)

Dataset

Example Data Instance:

track_id: 5SuOikwiRyPMVoIQDJUgSV
artists: Gen Hoshino
track_name: Comedy
popularity: 73
danceability: 0.676
energy: 0.461
acousticness: 0.0322
valence: 0.715
track_genre: acoustic

Statistics:

Thousands of songs with audio features from Spotify API

13+ audio features per song (danceability, energy, acousticness, etc.)

Includes metadata like artist, track name, album, and genre

Source: Spotify Web API / Kaggle dataset

Method/Technique

Core Components:

1 Data Preprocessing:

Normalization of audio features

2 K-means Clustering

Group similar songs into clusters

3 Cosine Similarity:

Calculate similarity between user's songs and potential recommendations

4 Weighted Profile Creation:

Allow prioritization of certain songs

Pipeline:

User Input → Data Preprocessing → Feature Extraction → Similarity Calculation → Recommendation Generation

Method: Technical Details

Clustering Algorithm:

- K-means with 10 clusters
- MinMaxScaler for feature normalization

Similarity Calculation:

```
def cosine_similarity(vec1, vec2):
    dot_product = np.dot(vec1, vec2)
    magnitude1 = np.linalg.norm(vec1)
    magnitude2 = np.linalg.norm(vec2)
    similarity = dot_product / (magnitude1 * magnitude2)
    return similarity
```

Weighted Profile:

- Create user profile as weighted average of song features
- Allow different importance for each song in history

Results

Performance Metrics:

Recommendation Relevance: High similarity scores (>0.85) for recommended songs
Cluster Cohesion: Similar audio characteristics within clusters
System Response Time: < 3 seconds for recommendations (dataset dependent)

Evaluation Method:

Cosine similarity between user profile and recommendations
Cluster analysis for genre and audio feature consistency

Analysis

Key Insights:

- Songs cluster primarily based on energy, danceability, and acousticness
- Genre patterns emerge naturally from audio features
- User preferences can be effectively captured through weighted profiles
- Songs with similar audio features often belong to related genres

Pattern Example:

- Acoustic songs typically have high acousticness (>0.7) and low energy (<0.4)Consider various channels such as online platforms, partnerships, and offline marketing.
- Dance songs show high danceability (>0.7) and energy (>0.6)

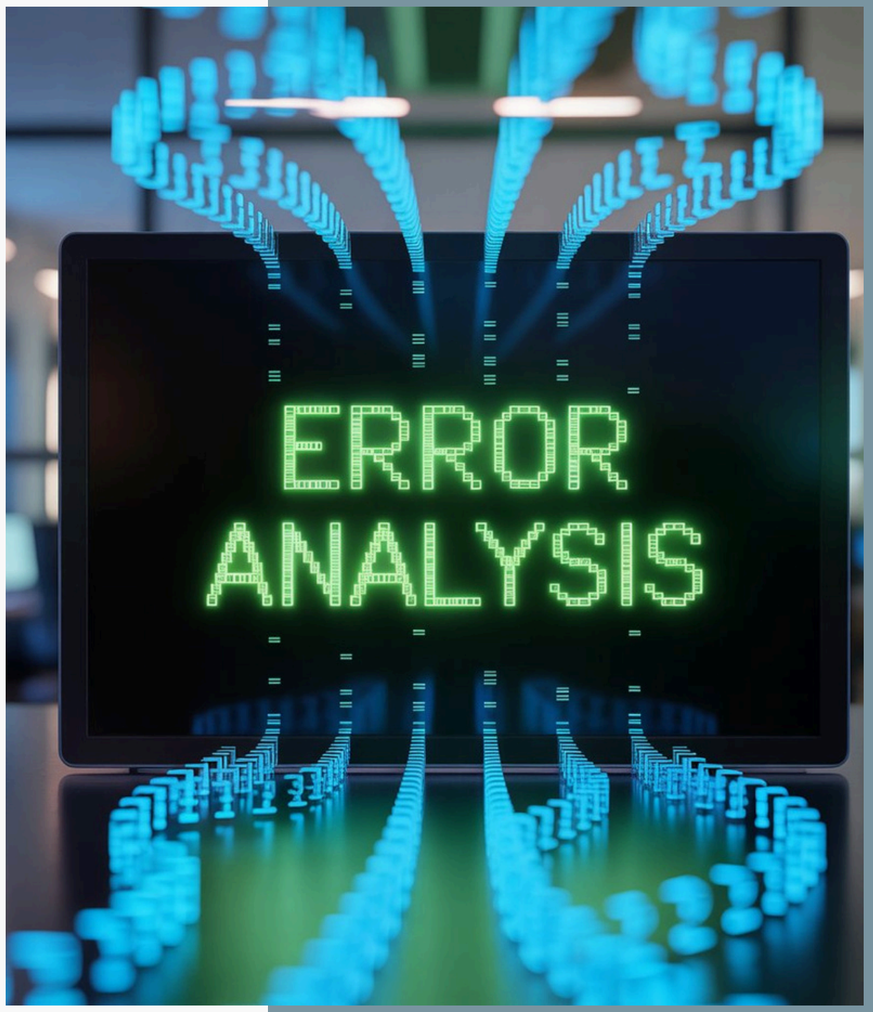
Error Analysis

Common Failure Cases:

- Obscure songs not found in the dataset
- Genre misclassification in recommendations
- Difficulty handling niche musical preferences

Error Sources:

- Limited dataset size compared to commercial systems.
- K-means limitations with non-spherical clusters
- Feature weighting might overemphasize certain audio characteristics



Improvements over Base Methodology

Added Weighted Recommendation:

- Enhanced personalization by allowing song weighting

Cluster Filtering:

- Improved recommendation relevance using cluster-based filtering

Interactive Visualization

- Added cluster analysis visualizations for better understanding

Progress Feedback:

- Implemented progress bars for better user experience

Multi-song Input:

- Support for multiple reference songs versus single-song recommendations

Learnings

Technical:

1. Applied clustering and similarity algorithms to real-world data
2. Implemented recommendation systems using content-based filtering
3. Developed data preprocessing pipeline for audio features

Problem-solving:

1. Identified optimal number of clusters for the dataset
2. Designed effective similarity metrics for music recommendation
3. Balanced computation efficiency with recommendation quality

Demo: Streamlit Interface

Key Components:

- Basic Recommendations: Get recommendations based on song history
- Weighted Recommendations: Assign importance to different songs
- Cluster Analysis: Explore song clusters and their characteristics
- About: System information and dataset statistics

The system features a responsive web interface with interactive elements and visualizations.

Summary and Conclusion

Project Recap:

- Built a Spotify song recommendation system using K-means clustering and cosine similarity
- Implemented basic and weighted recommendation modes
- Created cluster analysis tools for music exploration

Future Work:

- Incorporate user feedback and listening history
- Integrate with Spotify API for real-time recommendations
- Expand dataset size and diversity
- Experiment with advanced algorithms (SVD, neural networks)

Thank you