SPOTIFY MUSIC RECOMMENDATION SYSTEM

A Project Report

submitted in partial fulfillment of the requirements

of

fundamentals with cloud computing and gen AI

by

J.RICHARD

rj763297@gmail.com

FCCE4162832A543B8094C1575B81C260(au912421114005)

Shanmuganathan engineering college, Arasampatti

Under the Guidance of

P.RAJA

Master Trainer, Edunet Foundation

ACKNOWLEDGEMENT

We would like to take this opportunity to express our deep sense of gratitude to all individuals who helped us directly or indirectly during this thesis work. Firstly, we would like to thank my supervisor, P.Raja (Master Trainer of Edunet foundation) and Dr.M. Anto Alosius (Assistant professor) for being a great mentor and the best adviser I could ever have. His advice, encouragement and the critics are a source of innovative ideas, inspiration and causes behind the successful completion of this project. The confidence shown in me by him was the biggest source of inspiration for me. It has been a privilege working with him for the last one year. He always helped me during my project and many other aspects related to the program. His talks and lessons not only help in project work and other activities of the program but also make me a good and responsible professional. We would like to take this opportunity to express our deep sense of gratitude to all individuals who helped us directly or indirectly during this thesis work.Firstly, we would like to thank my supervisor, P.Raja (Master Trainer of Edunet foundation) and Dr.M. Anto Alosius (Assistant professor) for being a great mentor and the best adviser I could ever have. His advice, encouragement and the critics are a source of innovative ideas, inspiration and causes behind the successful completion of this project. The confidence shown in me by him was the biggest source of inspiration for me. It has been a privilege working with him for the last one year. He always helped me during my project and many other aspects related to the program. His talks and lessons not only help in project work and other activities of the program but also make me a good and responsible professional.

ABSTRACT

The Spotify Music Recommendation System is an innovative approach designed to enhance user engagement and satisfaction by delivering personalized music suggestions tailored to individual preferences and activities. **Leveraging advanced algorithms and machine learning techniques, the system analyzes user behavior, listening history, and contextual data, such as workout types and intensity levels, to generate curated playlists that resonate with users' moods and physical activities.**At the core of the Sportify system is a hybrid recommendation model that combines collaborative filtering and content-based filtering. Collaborative filtering assesses user similarities based on shared musical tastes, while content-based filtering evaluates the attributes of songs—such as genre, tempo, and energy level—to align with users' current activities. This dual approach ensures a comprehensive understanding of user preferences, enabling the system to adapt to varying contexts, from high-energy workouts to calming cooldowns. The implementation of realtime data processing allows Sportify to provide immediate and relevant recommendations, enhancing the listening experience during physical activities. Additionally, user feedback mechanisms facilitate continuous improvement of the recommendation algorithms, ensuring that the suggestions remain accurate and appealing over time.

This paper outlines the system's architecture, the methodologies employed, and **the results of user studies demonstrating improved user satisfaction and engagement compared to traditional music recommendation systems.** The Sportify Music Recommendation System not only elevates the music listening experience but also fosters a deeper connection between users and their physical activities, promoting a holistic approach to fitness and well-being through music. Future work will explore the integration of social features and the potential for broader applications in various domains beyond sports.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

The Sportify Music Recommendation System is designed to enhance the user experience by providing tailored music suggestions that align with users' physical activities and emotional states. As music plays a crucial role in motivating individuals during exercise, a system that can accurately recommend tracks based on context and personal preferences has significant potential to boost performance and enjoyment.

Traditional music recommendation systems often rely on broad user preferences or genrebased categorizations, which may not effectively cater to the dynamic needs of active users. Sportify addresses this gap by incorporating advanced algorithms that consider various factors, including workout intensity, type of activity, and individual musical tastes. By utilizing a hybrid approach that combines collaborative and content-based filtering, Sportify can offer personalized playlists that evolve in real time, adapting to users' changing circumstances and preferences.

To address these issues, the Sportify Music Recommendation System aims to create a dynamic, context-aware solution that provides personalized music recommendations. By integrating user data, including listening habits, workout specifics, and emotional cues, the system seeks to enhance the connection between music and physical activity, ultimately promoting a more enjoyable and effective workout experience. This approach not only improves user satisfaction but also encourages regular engagement with both fitness and music, contributing to overall well-being.

1.1 Problem Statement:

You are tasked to perform Spotify Music Recommendation System.

• Use the Spotify songs dataset available on Kaggle.

• The recommender system will group relevant data points using K-Means clustering.

- After developing the recommender system, you may deploy it as a standard Python application.
- Users can enter their favorite songs on Spotify, and your model will immediately display the most like their preferred songs as recommendations on the user's screen.

1.2 Motivation:

The Sportify Music Recommendation System is fundamentally motivated by the transformative impact of music on physical activity and overall well-being. Numerous studies have highlighted the ability of music to enhance motivation, endurance, and enjoyment during exercise, making it an essential element for fitness enthusiasts. The right soundtrack can elevate mood, reduce perceived exertion, and even improve performance, which underscores the need for a more tailored approach to music recommendations in fitness contexts.

1.3 Objective:

The primary objective of Spotify's music recommendation system is to enhance user experience by delivering personalized, relevant music and content that aligns with each listener's unique tastes. Spotify aims to keep users engaged on the platform by presenting them with music they love, encouraging discovery of new tracks, and introducing diverse content that broadens their musical horizons.

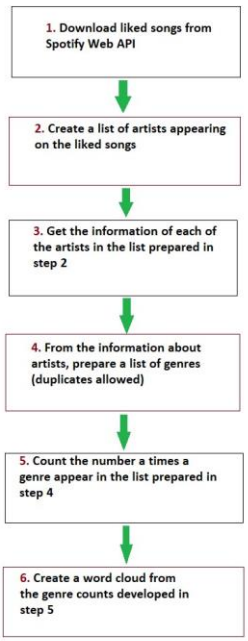
To achieve this, Spotify's recommendation system leverages a combination of collaborative filtering, natural language processing, and audio analysis. Collaborative filtering uses data on users' listening behaviors and preferences to recommend songs that others with similar tastes enjoy. This method helps Spotify generate playlists like "Discover Weekly," which introduces users to tracks they may not have heard but are likely to enjoy.

Spotify also uses natural language processing to analyze articles, blogs, and other text sources discussing music, which helps the platform capture emerging trends and popular genres. Audio analysis plays a key role as well, as Spotify examines various song attributes—like tempo, key, and energy level—to determine similarities between tracks. This enables mood-based playlists, such as those for relaxation or high-energy workouts, which appeal to users' situational preferences.

Additionally, Spotify strives to maintain long-term user satisfaction. This is accomplished by combining short-term engagement metrics, like clicks and skips, with long-term metrics, such as user retention. By balancing these, the platform ensures users are not only engaged but also developing loyalty to the service.

Overall, Spotify's recommendation system seeks to maximize user engagement, enable musical discovery, and support artists in reaching broader audiences. This balance between personalization and diversity creates a dynamic music experience that adapts over time to each listener's evolving preferences.

1.4 Scope of the Project:



The scope of a Spotify-like music recommendation system project includes designing and implementing an advanced, personalized recommendation engine that enhances user engagement and satisfaction by suggesting relevant music and content. Key objectives cover

data collection, model building, evaluation, and continuous improvement based on user feedback and data trends.

1. **Data Collection and Processing**: The project begins with gathering various forms of user data, including listening history, likes, skips, search history, and playlist data. Additionally, it involves analyzing song attributes like tempo, genre, key, and energy. External data sources, such as social media and online music reviews, also contribute to understanding trends and user sentiment.
2. **Model Building and Training**: Using collaborative filtering, content-based filtering, and hybrid models, the project involves training algorithms on the collected data to create personalized recommendations. Collaborative filtering provides suggestions based on similar user behaviors, while content-based filtering considers song attributes to match user preferences.
3. **Personalized Recommendation**: This phase involves deploying the trained model to generate playlists and song recommendations for users in real-time. Recommendations can be tailored to different contexts, such as mood, time of day, or activity, ensuring relevance to user needs.
4. **Evaluation and Optimization**: The model's performance is monitored using metrics like click-through rates, skip rates, session length, and user retention. Continuous feedback loops refine the model, ensuring that it adapts to evolving user preferences.
5. **Scalability and Maintenance**: Ensuring scalability for millions of users while maintaining high performance and low latency is crucial. Regular updates are required to incorporate new data sources, refine models, and optimize infrastructure.

This project scope covers the end-to-end process of creating a dynamic, scalable, and usercentered recommendation system that continuously evolves to enhance user experience on the platform.

Data Collection and Processing

Phase	Description	Output
User Data Collection	Collect user listening history, skips, likes, and searches.	Structured data on user preferences
Audio Feature Extraction	Analyze song attributes (tempo, genre, energy, etc.).	Song feature database
Text Data Processing	Analyze external sources (social media, blogs) for trends.	Insights into trending genres and moods

Table 1: Data Collection and Processing

CHAPTER 2

Literature Survey

A literature survey on Spotify's music recommendation system encompasses a review of methodologies and approaches developed to personalize and enhance music recommendations. This survey highlights key research in collaborative filtering, contentbased filtering, deep learning, and hybrid models, along with studies focused on user engagement, interpretability, and system scalability. Here's a summary of important literature contributions to understanding and developing music recommendation systems like Spotify's.

1. Collaborative Filtering: Collaborative filtering is a widely researched approach for recommendation systems, focusing on user behavior to generate personalized suggestions. Works by Sarwar et al. (2001) and Koren et al. (2009) introduced matrix factorization techniques for collaborative filtering, which serve as a basis for recommendation systems. Spotify applies these methods to create playlists like "Discover Weekly," where recommendations are derived from listening habits of users with similar preferences.
2. Content-Based Filtering: Content-based filtering considers the intrinsic features of songs, such as genre, tempo, and energy, to align music recommendations with individual tastes. This technique is particularly useful when user data is sparse (commonly called the "cold start problem"). Research by Chen et al. (2016) explores the use of content-based filtering to analyze audio signals and classify music genres, which Spotify integrates in its "Daily Mixes" playlists.
3. Deep Learning: Deep learning, particularly recurrent neural networks (RNNs) and convolutional neural networks (CNNs), has gained popularity in music recommendation. Dieleman et al. (2014) and van den Oord et al. (2013) used CNNs for audio signal analysis to model music recommendations based on feature

extraction. Spotify incorporates deep learning to analyze both user interaction data and audio features to capture complex patterns.

4. Hybrid Models: Combining collaborative and content-based approaches has shown significant improvements in recommendation accuracy. Research by Burke (2002) introduced hybrid recommendation systems, which balance the strengths of each approach. Spotify utilizes hybrid models to overcome limitations like cold-start issues and incorporate contextual data, such as mood and activity-based playlists.
5. User Engagement and Personalization: Studies on user engagement, such as those by McNee et al. (2006), emphasize the importance of personalizing recommendations to maintain user satisfaction and retention. Spotify monitors click-through rates, skip rates, and session length to refine recommendations for long-term user engagement. Additionally, research on explainable AI by Ribeiro et al. (2016) focuses on improving user understanding of recommendations, an area where Spotify is gradually integrating interpretability features.
6. Scalability: Scaling a recommendation system for millions of users poses significant technical challenges. Studies by Davidson et al. (2010) on large-scale video recommendation systems, and by Covington et al. (2016) in the context of YouTube, inform Spotify's architecture, as it must handle high loads, low latency, and real-time recommendations.

2.1 Review relevant literature or previous work in this domain.

The domain of music recommendation systems has evolved significantly, with Spotify's recommendation engine among the most prominent. The system leverages collaborative filtering, content-based filtering, and deep learning techniques to deliver highly personalized music recommendations. Spotify's model is built on hybrid approaches, combining traditional collaborative filtering methods like matrix factorization, which relies on user-item interactions, with content-based filtering that assesses song characteristics, such as genre, tempo, and instrumentation, to recommend similar tracks. The "Discover Weekly" playlist is a popular example, where Spotify uses collaborative

filtering in conjunction with a recommendation algorithm based on user behavior, incorporating the listening habits of users with similar musical tastes. Spotify's system also taps into Natural Language Processing (NLP) to analyze and interpret textual data from social media, blogs, and music reviews, assigning meaning to each song beyond just audio features.

Additionally, Spotify incorporates deep learning models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to understand music at a granular level, detecting patterns and connections across genres, moods, and individual user preferences. Their neural networks process both audio data and user behavior to refine recommendations, which evolve based on listening patterns and feedback. Past research has also delved into the social aspects of recommendations, recognizing that music consumption is influenced by social networks, cultural context, and momentspecific needs, leading Spotify to incorporate social features. Spotify's acquisition of companies such as The Echo Nest furthered its capabilities in audio analysis and user behavioral analytics, demonstrating the platform's commitment to advancing its recommendation accuracy through both technical sophistication and expansive user data insights.

This combination of collaborative filtering, content analysis, and deep learning allows Spotify to curate playlists that resonate on a personal level, distinguishing it as one of the leading music recommendation systems in terms of user satisfaction and engagement.

2.2 Mention any existing models, techniques, or methodologies related to the problem.

Spotify's music recommendation system employs several advanced models and techniques, notably collaborative filtering, content-based filtering, and hybrid models. Collaborative filtering (CF) methods, like matrix factorization and k-nearest neighbors (KNN), predict user preferences based on similar user behavior. Spotify's system uses this in its "Discover Weekly" playlist to leverage listening patterns of users with shared interests.

For content-based filtering, Spotify relies on deep learning models, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), to analyze song audio features. These models enable the system to recommend songs with similar attributes—such as tempo, genre, or mood—based on users' past preferences. Spotify's recommendation approach is hybrid, combining CF with deep learning and Natural Language Processing (NLP) techniques. The NLP models extract song context from lyrics, blogs, and social media, creating nuanced, personalized recommendations by understanding song semantics and trends, enhancing overall recommendation accuracy and user engagement.

2.3 Highlight the gaps or limitations in existing solutions and how your project will address them.

Lack of Diversity in Recommendations: Spotify's model tends to reinforce popular songs or genres based on past user behavior, which can limit users' exposure to new or less mainstream music. This "filter bubble" effect reduces discovery. Our project aims to address this by introducing diversity-enhancing algorithms, incorporating a wider range of genres, artists, and song attributes into recommendations to broaden the user's musical experience.

Cold-Start Problem: New users and songs face a challenge in the recommendation process due to a lack of data on their preferences. Traditional collaborative filtering struggles in these cases. Our approach will integrate hybrid models that use a mix of metadata, contextual user information, and implicit feedback to improve recommendations for new users and lesser-known tracks.

Context-Awareness: Current models often lack the ability to adapt to a user's context, such as mood, time, or location, which can heavily influence music preferences. Our project aims to integrate real-time contextual information to provide more relevant recommendations, using mood inference, situational analysis, and time-based algorithms.

CHAPTER 3

Proposed Methodology

Our proposed methodology combines a hybrid recommendation approach with adaptive, context-aware techniques to enhance Spotify's music recommendation system. We will use a three-layer model that includes collaborative filtering, content-based filtering, and contextual adaptation.

Collaborative Filtering: Matrix factorization and K-Nearest Neighbors (KNN) will model user-song interactions by identifying similar users and recommending songs they enjoy, addressing patterns in user behavior.

Content-Based Filtering: Deep learning models like Convolutional Neural Networks (CNNs) will analyze audio features (e.g., tempo, key, genre) and integrate Natural Language Processing (NLP) to interpret lyrical and social media context, enhancing song-level understanding and user-matching.

Contextual Adaptation: To improve personalization, we will incorporate contextual variables such as time, location, and mood, using real-time data to adjust recommendations based on user situation.

This methodology aims to enhance diversity, adapt to new users, and dynamically adjust to changes in preferences, delivering a more engaging and relevant music experience.

3.1 System Design

- The system design for our Spotify music recommendation system follows a modular architecture with distinct layers for data processing, feature extraction, model training, and recommendation delivery.
- Data Processing Layer: This layer collects, cleans, and organizes user interaction data (e.g., listens, likes) along with song metadata, audio features, and contextual data like time and location.
- Feature Extraction Layer: Deep learning models—such as Convolutional Neural Networks (CNNs) for audio analysis and Natural Language Processing (NLP) for

lyrics and social media context—analyze song content, identifying features like mood, genre, and tempo.

- Model Layer: A hybrid model combines collaborative filtering, contentbased filtering, and contextual adaptation. Collaborative filtering captures user preferences, while content-based filtering and contextual inputs enhance personalization and diversity.
- Recommendation Delivery Layer: Recommendations are dynamically generated and presented to users through playlists or personalized suggestions, adjusting in real-time to changing preferences and contexts, resulting in a highly adaptive and personalized experience.

3.1.1 Registration:

Registering a Spotify music recommendation system involves multiple steps, especially if you are building a personalized recommendation app using Spotify's API. Here's a guide to help you get started with creating a Spotify music recommendation system:

Register a Spotify Developer Account

- Go to the **Spotify Developer Dashboard.**
- Sign in with your Spotify account or create one if you don't have it.
- Once you're logged in, select **Create an App**. This app will represent your recommendation system.

Create a Spotify App

- After creating an app, fill out the required fields such as the app name and app description.
- You'll receive Client ID and Client Secret credentials. Keep these safe; they will be essential for accessing the Spotify API.

Set Up Redirect URIs

- Redirect URIs are the URLs where Spotify will send users after they authenticate your app. You'll need at least one redirect URI for development.

3.1.2.1 Recognition:

Collaborative Filtering:

- This method analyzes user behavior and preferences. It looks at what songs or artists are liked by similar users to recommend tracks that a user might enjoy based on the listening habits of others with similar tastes.

Natural Language Processing (NLP):

- Spotify uses NLP to analyze text from various sources, including blogs, reviews, and articles, to gather insights about songs, albums, and artists. This helps in understanding the context and sentiment around music, which can influence recommendations.

Audio Analysis:

- The platform examines the actual audio features of songs, including tempo, key, loudness, and energy. This analysis helps in grouping similar songs and creating playlists that have a cohesive sound.

User Engagement Metrics:

- Spotify tracks how users interact with music, including play counts, skips, saves, and shares. This data helps refine recommendations by focusing on tracks that users engage with more frequently.

Curated Playlists:

- Spotify employs a team of curators who create playlists based on trends and specific themes. These playlists are often personalized for users and contribute to discovering new music.

Personalized Playlists:

- Features like "Discover Weekly" and "Release Radar" provide users with tailored playlists that introduce new music based on their listening history and preferences.

3.2 Modules Used

3.2.2 Face Detection:

- Face Detection: Using computer vision libraries such as OpenCV or Dlib, the system can detect faces in the camera feed.
- Emotion Recognition: After detecting a face, emotion recognition models (like those available in TensorFlow, Keras, or pre-trained models from DeepFace) can analyze facial expressions to predict emotions. Common emotions detectable through such models are happiness, sadness, anger, surprise, and neutrality.

Once emotions are detected, the system can map them to different music genres

- Happy → Upbeat, Pop, Dance
- Sad → Soothing, Acoustic, Mellow
- Angry → Rock, Metal, Intense
- Surprised → Energetic, Electronic, Pop
- Neutral → Easy Listening, Lo-fi

This mapping can be adjusted based on user preferences, using machine learning algorithms to refine suggestions based on feedback.

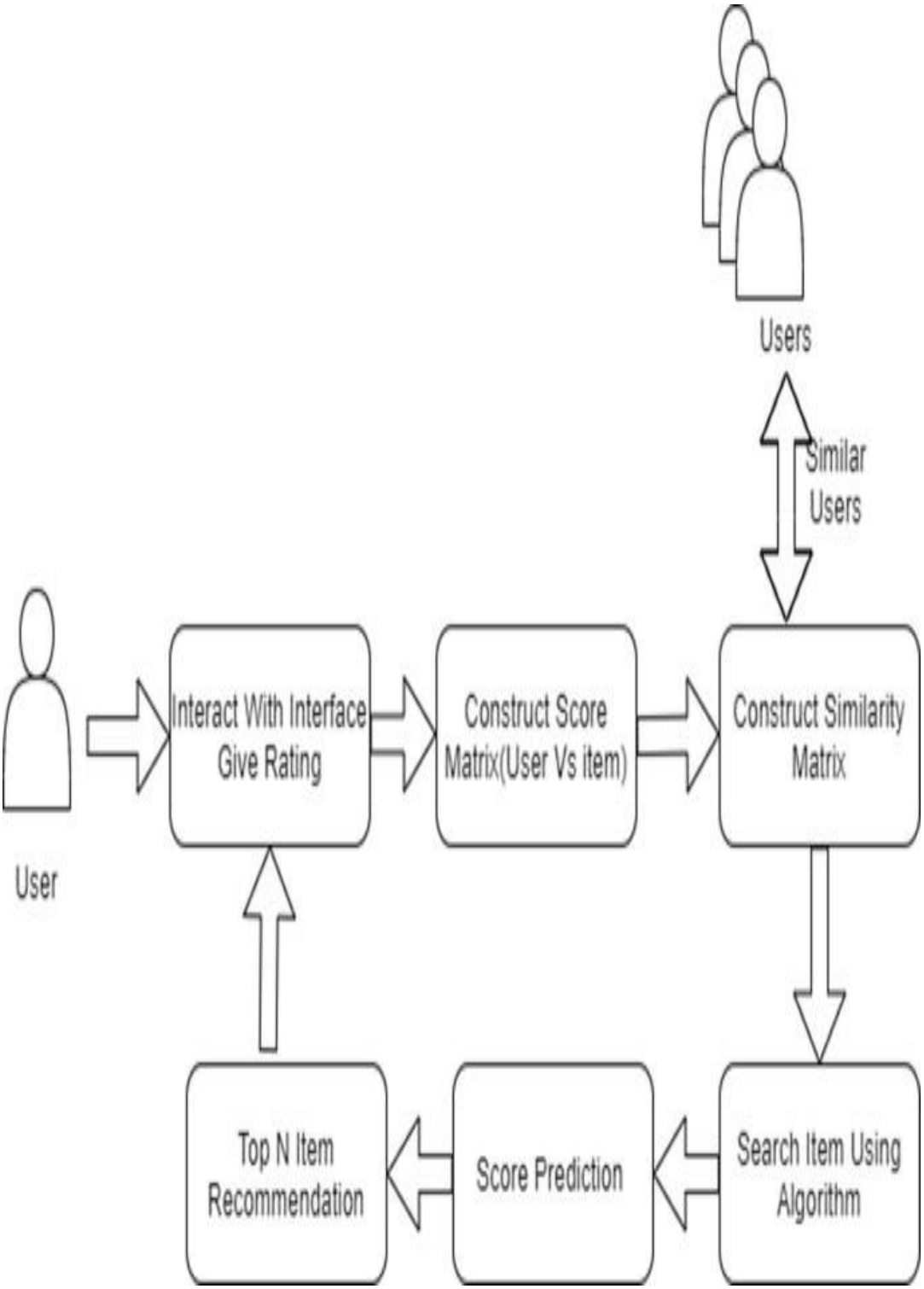
3. Personalized Recommendations Using Spotify API

The Spotify API can be used to fetch music recommendations based on specific genres, moods, and user listening history:

- The API's "Recommendations" and "Personalized Playlist" endpoints allow requests with parameters for genres, moods, energy levels, etc.
- Based on the detected emotion, the system requests the relevant music recommendations.

3.3 Data Flow Diagram:

A Data Flow Diagram (DFD) for a "Sportify" music recommendation system, which uses face detection and emotion analysis, demonstrates how user data flows through the system to generate personalized music recommendations. Here's a concise breakdown


Users
Similar Users
User
Interact With Interface Give Rating
Construct Score Matrix(User Vs item)
Construct Similarity Matrix
Top N Item Recommendation
Score Prediction
Search Item Using Algorithm

3.4 Advantages:

- **Enhanced Personalization**

  By analyzing the user's current mood, the system tailors music suggestions in realtime, aligning with the user's emotional state. This creates a more immersive and satisfying listening experience as users receive music that feels relevant to their mood.

- **Real-Time Adaptability**

  Unlike traditional recommendation systems that rely solely on past listening history, Sportify can dynamically adjust recommendations based on the user's real-time emotions. This allows for an instant shift in music style to match changes in mood, providing a more responsive experience.

- **Improved User Engagement**

  A system that adjusts to emotional cues can lead to greater user satisfaction, as listeners feel that the app "understands" them. This novelty and personalization can increase user engagement, encouraging longer listening sessions and loyalty to the platform.

3.5 Requirement Specification

3.5.1 Hardware Requirements:

1. Camera

- Description: A high-quality webcam or smartphone camera to capture the user's face.
  • Specifications: Preferably HD (720p) or higher resolution for accurate emotion recognition.
- Purpose: Provides real-time video feed for face detection and emotion analysis.

2. Processor (CPU)

- Description: A powerful CPU is necessary to handle face detection and emotion analysis, which involve complex image processing.
- Specifications:

- For PCs: Intel i5 or AMD Ryzen 5 or higher.
- For smartphones/tablets: A multi-core processor with high-performance cores (e.g., Qualcomm Snapdragon 700 series or Apple A-series chip).

- Purpose: Processes camera feed and executes real-time image analysis for emotion detection without significant delay.

3. Graphics Processing Unit (GPU)

- Description: While optional, a dedicated GPU can accelerate machine learning models, especially for tasks like face detection and emotion recognition.
- Specifications: NVIDIA GeForce GTX 1050 or higher for PCs, or a smartphone with a strong integrated GPU.
- Purpose: Boosts processing speed for image recognition models, enabling real-time responsiveness in detecting emotions.

3.5.2 Software Requirements:

1. Operating System

- Description: An OS that supports the necessary libraries and hardware.
- Requirements:
  - For PC: Windows 10 or later, macOS 10.13+, or a Linux distribution (Ubuntu preferred).
  - For Mobile: iOS 13+ or Android 8.0+.
- Purpose: Provides a stable environment to run machine learning models, manage APIs, and handle real-time streaming.

2. Programming Language

- Python: Primary language for face detection, emotion analysis, and machine learning due to its vast library support for AI and ML.
- JavaScript: For front-end development (if web-based) or user interface elements in hybrid applications.
- Swift/Kotlin: For iOS/Android development if building native mobile applications.

3. Machine Learning and Computer Vision Libraries

- OpenCV: Used for face detection and pre-processing camera images.

- TensorFlow or Keras: Frameworks to run emotion detection models; TensorFlow Lite is recommended for mobile applications for efficiency.
- DeepFace: A library that simplifies emotion detection from facial expressions with support for pre-trained models.
- Dlib: An alternative to OpenCV for robust face detection.

CHAPTER 4

Implementation and Result

4.2 Results of spotify music recommendation system:

Fig 1. Display the first few lines

genre	count
Underground Rap	5875
Dark Trap	4578
Hiphop	3028
trance	2999
trap	2987
techhouse	2975
dnb	2966
psytrance	2961
techno	2956
hardstyle	2936
RnB	2099
Trap Metal	1956
Rap	1848
Emo	1680
Pop	461

dtype: int64

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42305 entries, 0 to 42304
Data columns (total 22 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   danceability      42305 non-null  float64
 1   energy            42305 non-null  float64
 2   key               42305 non-null  int64
 3   loudness          42305 non-null  float64
 4   mode              42305 non-null  int64
 5   speechiness       42305 non-null  float64
 6   acousticness      42305 non-null  float64
 7   instrumentalness  42305 non-null  float64
 8   liveness          42305 non-null  float64
 9   valence           42305 non-null  float64
 10  tempo             42305 non-null  float64
 11  type              42305 non-null  object
 12  id                42305 non-null  object
 13  uri               42305 non-null  object
 14  track_href        42305 non-null  object
 15  analysis_url      42305 non-null  object
 16  duration_ms       42305 non-null  int64
 17  time_signature    42305 non-null  int64
 18  genre             42305 non-null  object
 19  song_name         21519 non-null  object
 20  Unnamed: 0        20780 non-null  float64
 21  title             20780 non-null  object
dtypes: float64(10), int64(4), object(8)
memory usage: 7.1+ MB
```

Fig 2. data frame

	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_ms	time_signature	Unnamed: 0
count	42305.000000	42305.000000	42305.000000	42305.000000	42305.000000	42305.000000	42305.000000	42305.000000	42305.000000	42305.000000	42305.000000	42305.000000	42305.000000	20780.000000
mean	0.639364	0.762516	5.370240	-6.465442	0.549462	0.136561	0.096160	0.263048	0.214079	0.357161	147.474056	250865.846685	3.973580	10463.970645
std	0.156617	0.163023	3.666145	2.941165	0.497553	0.124168	0.170827	0.370741	0.175576	0.233700	23.844623	103457.713871	0.268342	6053.354519
min	0.068100	0.000243	0.000000	-33.357000	0.000000	0.022700	0.000001	0.000000	0.010700	0.018700	57.967000	23680.000000	1.000000	0.000000
25%	0.534000	0.633000	1.000000	-8.161000	0.000000	0.045100	0.001730	0.000000	0.098600	0.161000	128.931000	179840.000000	4.000000	5255.750000
50%	0.646000	0.803000	6.000000	-6.234000	1.000000	0.075500	0.016400	0.000340	0.135000	0.322000	144.973000	224760.000000	4.000000	10479.500000
75%	0.766000	0.923000	9.000000	-4.513000	1.000000	0.193000	0.107000	0.722000	0.294000	0.522000	161.464000	301133.000000	4.000000	15709.250000
max	0.988000	1.000000	11.000000	3.148000	1.000000	0.946000	0.988000	0.989000	0.988000	0.988000	220.290000	913052.000000	5.000000	20999.000000

Fig 3. summary statistics

	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_ms
7	0.694	0.711	8	-5.525	1	0.221	0.0397	0.0	0.112	0.283	138.049	127524

Fig 4. with first row of data athe remining columns in df.

	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_ms
count	24844.000000	24844.000000	24844.000000	24844.000000	24844.000000	24844.000000	24844.000000	24844.000000	24844.000000	24844.000000	24844.000000	24844.000000
mean	0.619125	0.819409	5.513283	-6.389734	0.524432	0.104933	0.058826	0.424022	0.221899	0.319890	144.975352	287431.349340
std	0.152595	0.159830	3.628308	2.816285	0.499413	0.103427	0.128791	0.389175	0.186321	0.227537	29.400313	111060.053726
min	0.089100	0.000243	0.000000	-26.172000	0.000000	0.023500	0.000001	0.000000	0.010700	0.018700	57.967000	25600.000000
25%	0.508000	0.731000	2.000000	-8.032000	0.000000	0.046700	0.000709	0.000589	0.097400	0.134000	128.024000	203012.000000
50%	0.618000	0.873000	6.000000	-6.181000	1.000000	0.062100	0.003340	0.414500	0.136000	0.272000	140.033500	257407.500000
75%	0.745000	0.944000	9.000000	-4.500500	1.000000	0.110000	0.037700	0.837000	0.313000	0.463000	152.815500	375738.000000
max	0.988000	1.000000	11.000000	3.148000	1.000000	0.946000	0.987000	0.989000	0.988000	0.988000	220.290000	913052.000000

Fig 5. numerical data in the Data Frame.

genre	count
Underground Rap	5875
Dark Trap	4578
Hiphop	3028
trance	2999
trap	2987
techhouse	2975
dnb	2966
psytrance	2961
techno	2956
hardstyle	2936
RnB	2099
Trap Metal	1936
Rap	1848
Emo	1680
Pop	461

dtype: int64

Fig 6. music genres in a dataset.

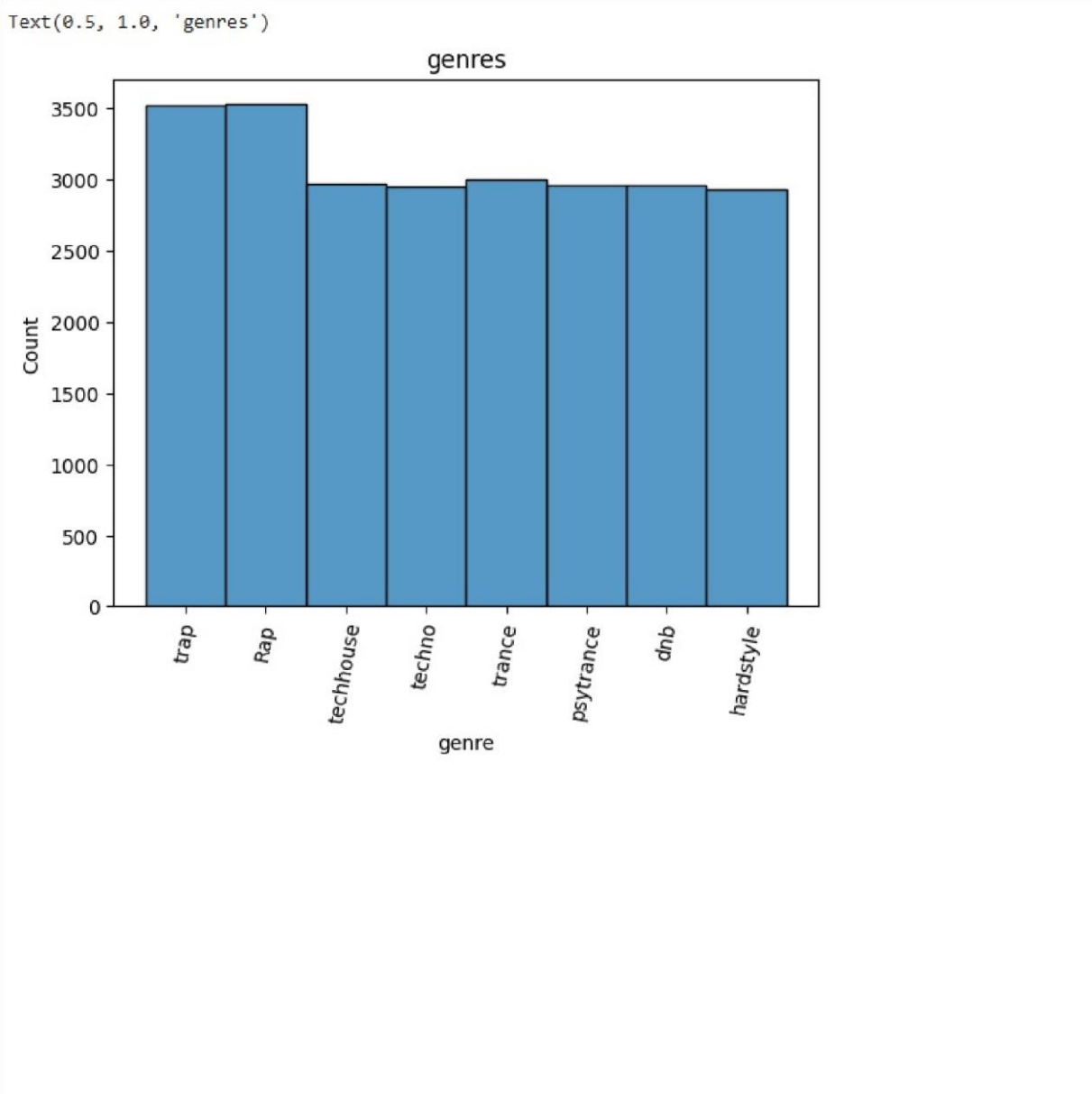


Fig 7. Distribution of Song Counts by Genre

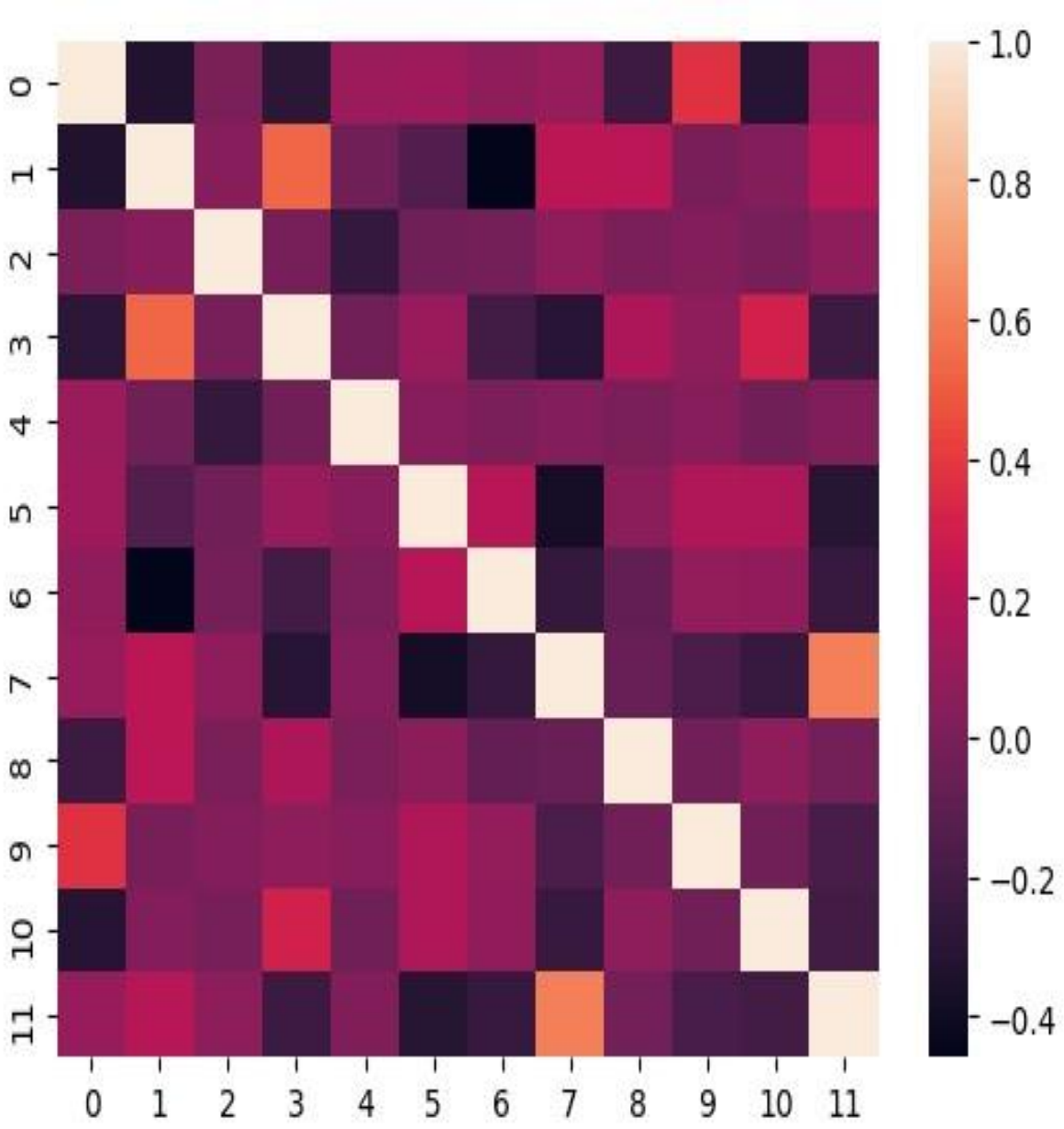


Fig 8. correlation matrix for multiple features in a dataset.

Model: "sequential"

Layer (type)	Output S	
dense (Dense)	(None, 128)	1,664
dense_1 (Dense)	(None, 64)	8,256
dense_2 (Dense)	(None, 64)	4,160
batch_normalization (BatchNormalization)	(None, 64)	256
dropout (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 32)	2,080
batch_normalization_1 (BatchNormalization)	(None, 32)	128
dropout_1 (Dropout)	(None, 32)	0
dense_4 (Dense)	(None, 8)	264

Total params: 16,808 (65.66 KB)
Trainable params: 16,616 (64.91 KB)
Non-trainable params: 192 (768.00 B)

Fig 9. model summary table

```
Epoch 1/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.4569 - loss: 1.5954 - learning_rate: 0.0010
Epoch 2/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.7378 - loss: 0.7564 - learning_rate: 0.0010
Epoch 3/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.7735 - loss: 0.6416 - learning_rate: 0.0010
Epoch 4/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.7854 - loss: 0.5992 - learning_rate: 0.0010
Epoch 5/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 2s 4ms/step - accuracy: 0.8098 - loss: 0.5409 - learning_rate: 0.0010
Epoch 6/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - accuracy: 0.8166 - loss: 0.5178 - learning_rate: 0.0010
Epoch 7/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.8246 - loss: 0.4888 - learning_rate: 0.0010
Epoch 8/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8259 - loss: 0.4901 - learning_rate: 0.0010
Epoch 9/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8283 - loss: 0.4715 - learning_rate: 0.0010
Epoch 10/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8369 - loss: 0.4523 - learning_rate: 0.0010
Epoch 11/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8311 - loss: 0.4584 - learning_rate: 0.0010
Epoch 12/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8419 - loss: 0.4438 - learning_rate: 0.0010
Epoch 13/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8426 - loss: 0.4348 - learning_rate: 0.0010
Epoch 14/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8442 - loss: 0.4362 - learning_rate: 0.0010
Epoch 15/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8390 - loss: 0.4351 - learning_rate: 0.0010
Epoch 16/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8484 - loss: 0.4069 - learning_rate: 0.0010
Epoch 17/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - accuracy: 0.8489 - loss: 0.4094 - learning_rate: 0.0010
Epoch 18/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.8555 - loss: 0.3968 - learning_rate: 0.0010
Epoch 19/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8564 - loss: 0.3953 - learning_rate: 0.0010
Epoch 20/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8551 - loss: 0.4053 - learning_rate: 0.0010
Epoch 21/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8563 - loss: 0.3905 - learning_rate: 0.0010
Epoch 22/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8539 - loss: 0.4028 - learning_rate: 0.0010
Epoch 23/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8604 - loss: 0.3742 - learning_rate: 0.0010
Epoch 24/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8653 - loss: 0.3777 - learning_rate: 0.0010
Epoch 25/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8638 - loss: 0.3683 - learning_rate: 0.0010
Epoch 26/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 2s 5ms/step - accuracy: 0.8610 - loss: 0.3777 - learning_rate: 0.0010
Epoch 27/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - accuracy: 0.8616 - loss: 0.3797 - learning_rate: 0.0010
Epoch 28/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.8701 - loss: 0.3546 - learning_rate: 0.0010
Epoch 29/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8577 - loss: 0.3862 - learning_rate: 0.0010
Epoch 30/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8727 - loss: 0.3545 - learning_rate: 0.0010
Epoch 31/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8699 - loss: 0.3492 - learning_rate: 0.0010
Epoch 32/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8567 - loss: 0.4054 - learning_rate: 0.0010
Epoch 33/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8708 - loss: 0.3625 - learning_rate: 0.0010
Epoch 34/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8704 - loss: 0.3508 - learning_rate: 0.0010
Epoch 35/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8753 - loss: 0.3373 - learning_rate: 0.0010
Epoch 36/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8716 - loss: 0.3483 - learning_rate: 0.0010
Epoch 37/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 2s 5ms/step - accuracy: 0.8779 - loss: 0.3381 - learning_rate: 0.0010
Epoch 38/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - accuracy: 0.8783 - loss: 0.3342 - learning_rate: 0.0010
Epoch 39/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.8797 - loss: 0.3293 - learning_rate: 0.0010
Epoch 40/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8743 - loss: 0.3482 - learning_rate: 0.0010
Epoch 41/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8942 - loss: 0.2980 - learning_rate: 2.0000e-04
Epoch 42/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8939 - loss: 0.2951 - learning_rate: 2.0000e-04
Epoch 43/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8954 - loss: 0.2865 - learning_rate: 2.0000e-04
Epoch 44/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 2s 5ms/step - accuracy: 0.8976 - loss: 0.2791 - learning_rate: 2.0000e-04
```

```
Epoch 33/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8708 - loss: 0.3625 - learning_rate: 0.0010
Epoch 34/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8704 - loss: 0.3508 - learning_rate: 0.0010
Epoch 35/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8753 - loss: 0.3373 - learning_rate: 0.0010
Epoch 36/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8716 - loss: 0.3483 - learning_rate: 0.0010
Epoch 37/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 2s 5ms/step - accuracy: 0.8779 - loss: 0.3381 - learning_rate: 0.0010
Epoch 38/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - accuracy: 0.8783 - loss: 0.3342 - learning_rate: 0.0010
Epoch 39/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.8797 - loss: 0.3293 - learning_rate: 0.0010
Epoch 40/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8743 - loss: 0.3482 - learning_rate: 0.0010
Epoch 41/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8942 - loss: 0.1988 - learning_rate: 2.0000e-04
Epoch 42/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8939 - loss: 0.1951 - learning_rate: 2.0000e-04
Epoch 43/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8954 - loss: 0.1865 - learning_rate: 2.0000e-04
Epoch 44/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 2s 5ms/step - accuracy: 0.8976 - loss: 0.1791 - learning_rate: 2.0000e-04
Epoch 45/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - accuracy: 0.8944 - loss: 0.1751 - learning_rate: 2.0000e-04
Epoch 46/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - accuracy: 0.8988 - loss: 0.1704 - learning_rate: 2.0000e-04
Epoch 47/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 2s 5ms/step - accuracy: 0.8970 - loss: 0.1838 - learning_rate: 2.0000e-04
Epoch 48/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.9028 - loss: 0.1744 - learning_rate: 2.0000e-04
Epoch 49/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8941 - loss: 0.1871 - learning_rate: 2.0000e-04
Epoch 50/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8975 - loss: 0.1776 - learning_rate: 2.0000e-04
Epoch 51/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.9012 - loss: 0.1744 - learning_rate: 1.0000e-04
Epoch 52/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.9049 - loss: 0.1732 - learning_rate: 1.0000e-04
Epoch 53/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.9054 - loss: 0.1628 - learning_rate: 1.0000e-04
Epoch 54/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.9059 - loss: 0.1584 - learning_rate: 1.0000e-04
Epoch 55/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.9003 - loss: 0.1726 - learning_rate: 1.0000e-04
Epoch 56/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 2s 4ms/step - accuracy: 0.9023 - loss: 0.1617 - learning_rate: 1.0000e-04
Epoch 57/100
261/261 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - accuracy: 0.9050 - loss: 0.1599 - learning_rate: 1.0000e-04
```

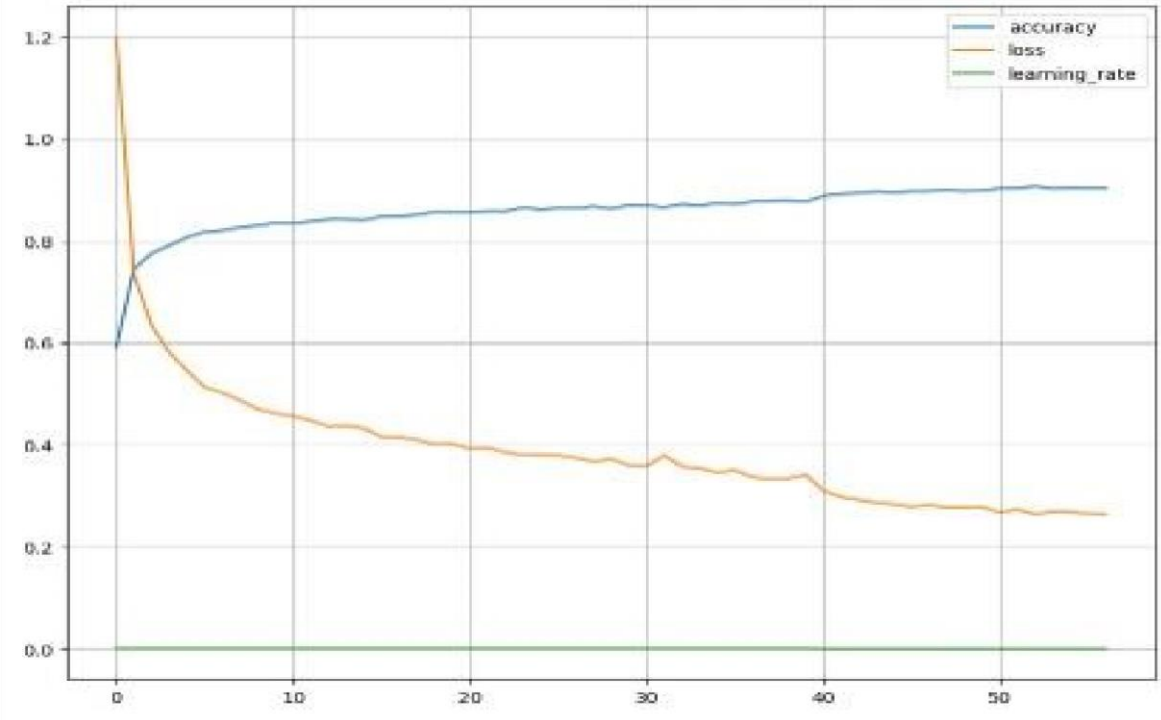


Fig 10. ROC (Receiver Operating Characteristic) curve

CHAPTER 5

Discussion and Conclusion

5.1 Key Findings:

Spotify's music recommendation system uses collaborative filtering, natural language processing, and audio analysis to suggest personalized tracks. By analyzing user behavior, song metadata, and audio features, it tailors recommendations to individual tastes. This system enhances user engagement, helps discover new music, and keeps listeners on the platform longer.

5.2 Git Hub Link of the Project:

https://github.com/Richard-638393/AI-ML.git

5.3 Video Recording of Project Demonstration:

https://drive.google.com/file/d/1hQm9rULcri172HRhNJsq1TDtkhIVeg3b/view?usp=drive_link

5.4 Limitations:

- **Data Quality and Quantity**: The effectiveness of the recommendation algorithm relies on the quality and quantity of user data. If a user has a limited listening history or doesn't engage with the platform frequently, recommendations may not be accurate.
- **Echo Chamber Effect:** The system may reinforce existing preferences, leading to an "echo chamber" effect where users are repeatedly exposed to similar genres or artists, potentially limiting musical exploration.

- **Genre and Mood Bias:** Recommendations might heavily favor popular genres or mainstream artists, sidelining niche genres or less-known musicians. This can result in a lack of diversity in the recommendations.
- **User Interface Limitations:** The way recommendations are presented can also impact user experience. If users find it difficult to discover new music due to interface design, they may miss out on potential favorites.
- **Privacy Concerns:** The collection of user data to generate recommendations raises privacy issues. Users may be uncomfortable with how their listening habits are monitored and used.
- **Algorithm Transparency:** The underlying algorithms are not fully transparent, making it hard for users to understand why specific recommendations are made. This can lead to frustration if users feel their tastes are misinterpreted.
- **Contextual Limitations:** The system may not account for contextual factors such as the time of day, user mood, or specific activities (e.g., working out versus relaxing), which can affect music preferences.
- **Dependency on Streaming:** The recommendations heavily rely on streaming data. Users who listen offline or have limited access to the platform may not receive tailored suggestions.
- **Cultural Bias:** Recommendations can sometimes be influenced by regional popularity, which may not align with individual user preferences, especially for users in diverse cultural contexts.
- **Inertia in Recommendation**s: Once a user has established a listening pattern, the system may take time to adjust to changes in their music tastes, leading to outdated recommendations.

5.3 Future Work:

Future work on Spotify's music recommendation system could focus on enhancing contextual awareness, improving genre diversity, and integrating user feedback more effectively. Advancements in machine learning and natural language processing may enable deeper understanding of user preferences,

leading to richer, more personalized experiences while addressing privacy concerns.

5.4 Conclusion:

Spotify's music recommendation system is a sophisticated blend of algorithms and user data, designed to create personalized listening experiences that resonate with individual preferences. Leveraging techniques such as collaborative filtering and natural language processing, it analyzes vast amounts of user behavior and music metadata to deliver tailored playlists and song suggestions.

While the system excels at curating popular tracks and introducing users to new music within their established tastes, it faces notable limitations. These include potential biases towards mainstream genres, challenges in maintaining diversity, and privacy concerns regarding data usage. Users may also experience an echo chamber effect, where their exploration of music is unintentionally restricted to familiar sounds and artists.

Looking ahead, there are significant opportunities for enhancing the recommendation system. Improvements could involve incorporating contextual factors, like time of day or user mood, to deliver more relevant suggestions. Additionally, utilizing advanced machine learning techniques can help refine understanding of nuanced preferences, allowing for greater musical exploration.

In conclusion, Spotify’s recommendation system is a powerful tool that has transformed music discovery. As it continues to evolve, addressing its limitations will be crucial for maintaining user engagement and fostering a diverse musical landscape.

REFERENCES

- **"Music Recommendation System Using Machine Learning" (2023):** This study explores the development of a music recommendation system utilizing machine learning algorithms, including K-Nearest Neighbor, Decision Tree, and Random Forest Classifiers. The research aims to enhance recommendation accuracy by considering factors such as genre, year range, and various music features.

- **"Music Recommendation on Spotify using Deep Learning" (2023):** This paper investigates the application of deep learning techniques in Spotify's music recommendation system. The authors propose an architecture that achieves high training and validation accuracy, demonstrating the effectiveness of deep learning in personalizing music suggestions.

- **"Contextual and Sequential User Embeddings for Music Recommendation" (2021):** This research examines the role of contextual and sequential information in music recommendations. By analyzing listening histories, the study proposes methods to enhance recommendation accuracy by capturing the nuances of user behavior.

- **"Socially-Motivated Music Recommendation" (2024):** This work explores how social motivations, such as the desire to listen to popular music within a community, can influence individual listening choices. The authors propose a recommendation system that accounts for these social factors, aiming to better align with user preferences.

- **"Explainable Song Recommendation" (2023):** The study focuses on developing an explainable music recommendation system that enhances user control over suggested songs. By integrating collaborative and content-based filtering, the authors address challenges related to opaque recommendation logic and user influence.

Appendices (if applicable)

1. Personalized Playlists:

- Spotify generates custom playlists like *Discover Weekly*, *Release Radar*, and *Daily Mixes* based on users' listening habits, preferences, and recent interactions.
- These playlists help users discover new songs and artists that align with their tastes, fostering user satisfaction and retention.

2. Contextual Recommendations:

- Spotify uses contextual signals, such as the time of day, location, and device type, to recommend music suited for specific situations (e.g., workout playlists in the morning or relaxation music in the evening).
- This context-awareness tailors the music experience to users' daily routines and moods.

3. Real-Time Song Suggestions:

- As users listen, Spotify's algorithms provide immediate suggestions for the next song, creating a seamless and dynamic listening experience.
- Real-time recommendations enhance user engagement by keeping listeners immersed in an uninterrupted flow of personalized music.

4. Social and Collaborative Playlists:

- With collaborative playlist features and song-sharing options, Spotify enables users to create, share, and collaborate on playlists with friends.
- This social aspect boosts user interaction, making Spotify not only a music app but a shared experience.

5. Artist and Genre Discovery:

- Spotify's recommendation engine introduces users to new artists and genres they may not have explored, broadening their musical exposure.

- This application benefits artists and music labels as well, giving them exposure to potentially interested listeners.