Music Recommendation System

Abstract

Music Recommendation system using NLP - Natural Language Processing using python language, which recommends music based on the content which includes:
1. Data Collection
2. Text Preprocessing
3. Feature Extraction
4. Building a Recommender Model & Content-Based Filtering
5. User Interaction and Recommendations

This essay/report will explore the implementation of a music recommendation system using NLP in Python, covering data collection, text preprocessing, feature extraction, building a recommender model, content-based filtering, and user interaction.

1. Introduction

The realm of music recommendation systems has gained significant relevance in tandem with the proliferation of music streaming platforms like Spotify and Tidal, enhancing accessibility to music for users far and wide. These systems play a pivotal role in uncovering new music gems by furnishing recommendations in the form of personally curated playlists that align with the user's musical tastes. Leading-edge systems currently rely on user-generated metadata, encompassing listening histories and prior purchases, to curate recommendations. Nonetheless, these metadata-centric systems encounter a hurdle termed the 'cold start' dilemma when confronted with insufficient data regarding novel songs or artists.
In response to this challenge, researchers have directed their efforts towards fortifying content-based recommender systems that leverage automatically extracted audio features inherent in the music content itself. The formulation of features and delineation of similarity within such systems are indispensable, albeit frequently executed ad-hoc and lacking optimization for the specific task at hand. The subjective nature of music similarity contributes to disparate interpretations driven by factors like genre, melody, rhythm, geographical origin, and instrumentation.
The intricacy of music recommendation is underscored by the multifaceted and subjective essence of music similarity, whereby assessments may diverge among listeners based on mood or evolve with time. This broad spectrum of viewpoints underscores the complexity in crafting an efficient recommendation system that is both challenging yet crucial for dispensing accurate suggestions to users.
The fusion of user interaction and real-time recommendations through a well-crafted user interface holds paramount importance in fostering user engagement and satisfaction with the system. Furthermore, conducting a comprehensive evaluation of the recommendation system is imperative for pinpointing areas ripe for enhancement and charting out future pathways for advancement. See references: [9] p. 6-10, [15], [4], [19], [23] p. 11-15.

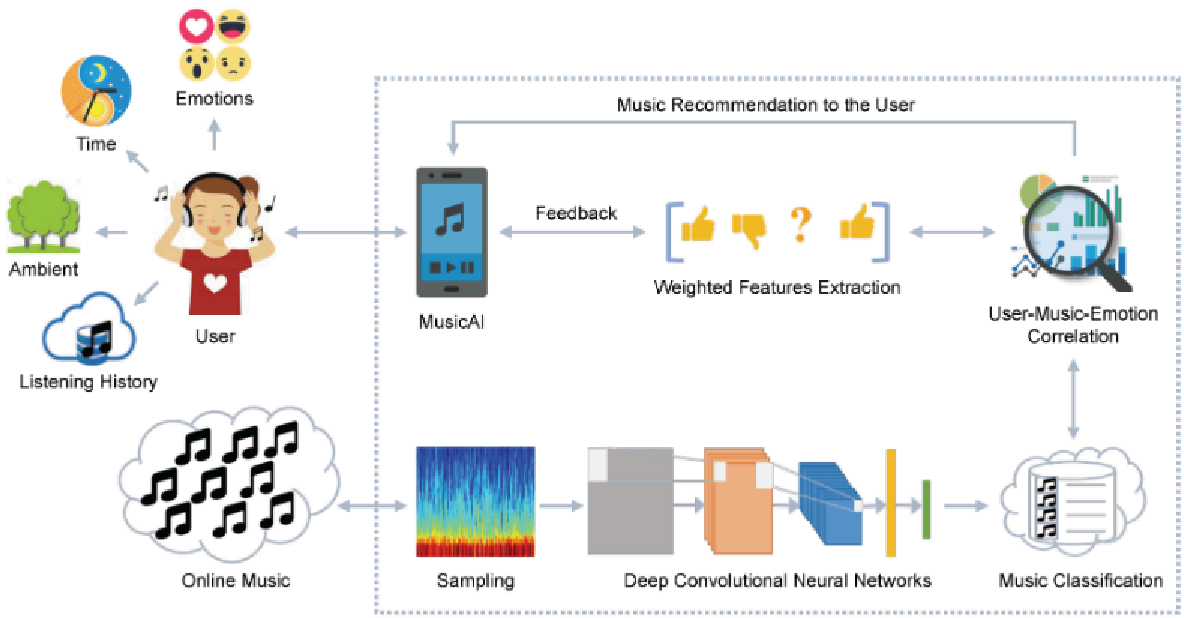

Figure 1: The system architecture of the EPMRS (emotion-aware personalized music recommendation system). (source: reference [3])

2. Background on Music Recommendation Systems

In this modern digital era, music stands out as a powerful form of expression, conveying emotions and ideas through auditory sensations. The rapid technological progress has led to the emergence of a wide array of music genres and styles, catering to diverse tastes across various cultures and regions. Consequently, there is a growing need for efficient music recommendation systems to tackle the overwhelming amount of data available in today's information-overloaded world.

Music recommendation systems play a vital role in sifting through extensive music libraries to extract relevant content based on user behaviors and preferences. Their primary objective is to offer tailored suggestions that align with individual interests, ultimately enhancing user satisfaction and involvement. Whether it involves curating personalized playlists on music streaming platforms or recommending similar tracks to boost sales on online music stores, these systems have become indispensable tools for elevating customer experiences and driving user engagement.

Key elements of a music recommendation system typically encompass user profiling, extraction of music features, recommendation algorithms, and techniques for matching users with relevant items. User profiles are built by taking into account factors such as age, gender, lifestyle, and hobbies to gain insights into their musical preferences. Music feature extraction entails analyzing metadata and acoustic attributes to enable personalized recommendations. Recommendation algorithms leverage collaborative filtering or content-based methods to propose music tracks that resonate with users' tastes.

The integration of advanced technologies like convolutional neural networks (CNNs) into music recommendation systems has significantly enhanced their accuracy and performance. CNNs can analyze users' listening habits and extract valuable information to predict their preferences accurately. Moreover, the emergence of emotion-aware personalized recommendation systems represents a groundbreaking approach to understanding users' emotional responses through their interactions with specific types of music.

As the demand for tailored recommendations continues to surge in the realm of digital music services, it is imperative for music recommendation systems to evolve continuously by embracing cutting-edge technologies such as CNNs and emotion-aware algorithms. By adopting innovative strategies in data collection, text preprocessing, feature extraction, and content-based filtering, these systems can deliver more precise and captivating recommendations that resonate with users' distinct tastes in an era characterized by abundant data resources. See references: [11], [16], [3].

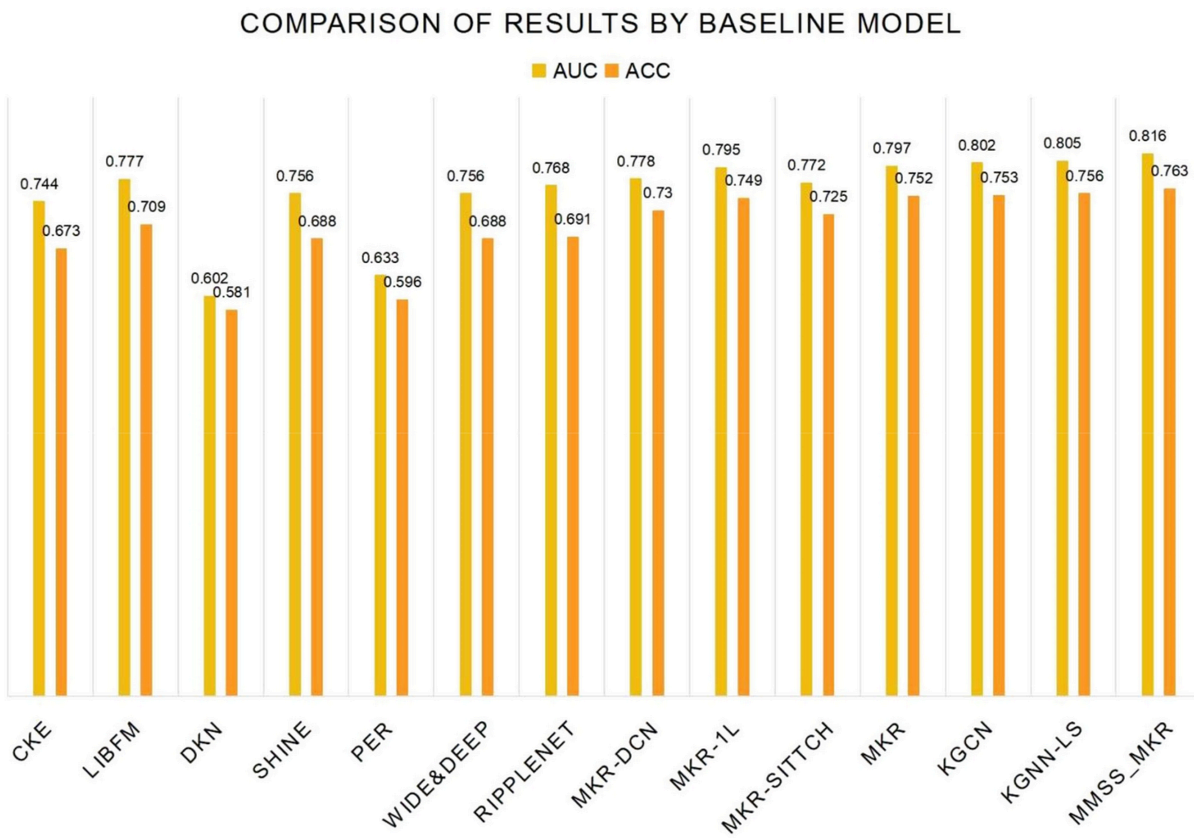

Figure 2: Histogram comparing results across baseline models. (source: reference [1])

Model	Last.FM	
AUC	ACC	
CKE	0.744	0.673
LibFM	0.777	0.709
DKN	0.602	0.581
SHINE	0.756	0.688
PER	0.633	0.596

Model	**Last.FM**	
AUC	**ACC**	
Wide&Deep	0.756	0.688
RippleNet	0.768	0.691
MKR-DCN	0.778	0.730
MKR-1L	0.795	0.749
MKR-sittch	0.772	0.725
MKR	0.797	0.752
KGCN	0.802	0.753
KGNN-LS	0.805	0.756
MMSS_MKR	**0.816**	**0.763**

Table 1: Comparison of modeling results for music datasets. (source: reference [1])

Model	**Book-crossing**	**MovieLens-1 M**		
AUC	**ACC**	**AUC**	**ACC**	
MKR	0.719	0.679	0.917	0.830
KGCN	0.684	0.641	0.915	0.839
KGNN-LS	0.721	0.681	0.918	0.840
MMSS_MKR	**0.731**	**0.690**	**0.921**	**0.843**

Table 2: Comparison of modeling results for book and movie datasets. (source: reference [1])

Dataset	**Haperparameters**
Last.FM	L = 1 H = 1 dim = 128 n_epochs = 200 Batch_size = 256 Lr_rs = $2\times {10}^{-5}$ Lr_kge = $2\times {10}^{-5}$ l2_weight = $2\times {10}^{-5}$ kge_interval = 64

Table 3: Hyperparameter settings. (source: reference [1])

Model	**Last.FM**	
AUC	**ACC**	
MKR	0.797	0.752
SM_MKR	0.809	0.759
MS_MKR	0.812	0.756
SS_MKR	0.811	0.760

Model	Last.FM	
AUC	ACC	
MMSS_MKR	0.816	0.763

Table 4: Ablation studies on the Last.FM dataset. (source: reference [1])

3. Data Collection

3.1. Web Scraping for Music Data

When it comes to music recommendation systems, the practice of web scraping for music data is crucial for enhancing the user experience by gathering valuable information. Numerous research studies have utilized a variety of datasets to train and evaluate their recommendation algorithms. For example, one study made use of the Last.FM dataset, which includes listening data from 2000 users of the Last.fm online music platform. This dataset covers a range of music details like artists, genres, media type, authors, and ratings.

In another study, researchers focused on an Emotion-Aware Personalized Music Recommendation System that utilized data from the MSD and JunoRecords websites. They selected songs from nine different genres and associated user information with music data to train their recommendation models. The accuracy of song recommendations was then tested across various genres for 100 users using different algorithms.

Additionally, user models for culture-aware music recommendation emphasized the significance of cultural and socio-economic cues in understanding music preferences. The dataset used in this particular study excluded certain tracks and listening events to create a dataset with 55,149 users, 394,944,868 listening events, and 3,478,399 distinct tracks. The average number of listening events per user was calculated to be 7,161, indicating a wealth of individual musical preferences.

Moreover, exploring correlations between cultural backgrounds and music preferences revealed potential links between acoustic properties of music and cultural or socio-economic characteristics. By examining relationships between various acoustic features and cultural/socio-economic dimensions, the study aimed to uncover insights into how these factors impact music preferences.

Overall, web scraping for music data is essential for collecting diverse and comprehensive datasets that are crucial for developing effective recommendation systems. By incorporating various sources of music data and understanding user preferences based on cultural signals and acoustic attributes, developers can create more personalized and engaging music recommendation experiences for listeners. See references: [6], [1], [3].

Dataset	#users#	#items#	#interactions	#KGtriples
Last.FM	1872	3846	42,346	15,518
Book-Crossing	17,860	14,910	139,746	19,793
MovieLens-1 M	6036	2347	753,772	20,195

Table 5: DataSets. (source: reference [1])

Genre	EPMRS	CSMRS	PMRSE
Breakbeat	74.96	60.66	52.33
Dancehall	94.56	80.44	73.15
Downtempo	76.52	61.91	50.98
Drum and bass	85.56	80.43	75.32
Funky house	79.55	82.45	74.53
Hip Hop	84.56	81.12	78.84
Minimal house	69.32	73.56	62.37
Rock	95.36	90.56	68.04
Trance	85.28	75.89	65.74

Table 6: The accuracy (%) of song recommendations in the nine genres for the 100 users. (source: reference [3])

Item	Value
Listening events	394,944,868
Users	55,149
Distinct tracks	3,478,399
Min. LE per user	1
Q_1 LE per user	1,442
Median LE per user	5,667
Q_3 LE per user	9,738
Max. LE per user	399,210
Avg. LE per User	7,161.41 (±10,326.91)
Avg. Users per Country	1,155.93 (±1,894.96)

Table 7: Statistics of the dataset utilized (LE = listening event). (source: reference [6])

Abbrv.	Country	Users
US	United States	10,251
RU	Russian Federation	5,021
DE	Germany	4,576
UK	United Kingdom	4,533
PL	Poland	4,403
BR	Brazil	3,882
FI	Finland	1,409
NL	Netherlands	1,375
ES	Spain	1,242
SE	Sweden	1,230
UA	Ukraine	1,140
CA	Canada	1,077
FR	France	1,055
AU	Australia	976
IT	Italy	973
JP	Japan	798
NO	Norway	750
MX	Mexico	705
CZ	Czechia	632
BY	Belarus	558
BE	Belgium	513
ID	Indonesia	484
TR	Turkey	478
CL	Chile	425
HR	Croatia	372
PT	Portugal	291
AR	Argentina	282
CH	Switzerland	277
AT	Austria	276
HU	Hungary	272
DK	Denmark	271
RS	Serbia	253

Abbrv.	Country	Users
RO	Romania	237
BG	Bulgaria	236
IE	Ireland	219
LT	Lithuania	202

Table 8: Number of users per country for countries with more than 200 users. We use ISO 3166 2-digit country codes to abbreviate country names. (source: reference [6])

	PD	IDV	MAS	UA	LTO	IND
Danceability	–0.035*	0.044*	0.023*	–0.052*	–0.024*	0.072*
Energy	0.056*	–0.102*	–0.014	**0.116***	0.076*	**–0.115***
Speechiness	0.022*	–0.034*	0.016*	0.085*	0.065*	–0.096*
Acousticness	–0.056*	**0.105***	0.026*	**–0.122***	–0.086*	**0.125***
Instrumentalness	–0.012	0.011	–0.029*	0.038*	0.055*	–0.055*
Liveness	0.021*	–0.042*	–0.014	0.059*	0.035*	–0.065*
Valence	–0.042*	0.059*	0.047*	–0.076*	–0.063*	**0.114***
Tempo	0.009	–0.041*	0.008	0.031*	0.043*	–0.025*

Table 9: Spearman rank-order correlations between users' acoustic properties of listening behavior and cultural features (Hofstede). Correlations >0.1 are highlighted in bold face. Statistically significant correlations at $p < 0.001$ are marked with an asterisk (*). (source: reference [6])

	Happiness	GDP	Social Sup.	Life Exp.	Freedom	Trust	Generosity
Danceability	0.035*	0.036*	–0.010	0.049*	0.037*	0.051*	0.052*
Energy	–0.036*	–0.067*	0.056*	–0.056*	–0.026*	–0.033*	**–0.101***
Speechiness	–0.018*	–0.007	0.059*	–0.017*	0.011	–0.004	–0.067*
Acousticness	0.055*	0.079*	–0.046*	0.070*	0.039*	0.048*	**0.118***
Instrumentalness	–0.031*	0.030*	0.042*	0.040*	0.006	0.001	–0.044*
Liveness	0.005	–0.019*	0.056*	–0.030*	0.001	–0.008	–0.048*
Valence	0.071*	0.047*	0.008	0.051*	0.044*	0.064*	0.084*

Tempo	0.004	–0.025*	0.046*	–0.015*	0.001	0.003	–0.016*

Table 10: Spearman rank-order correlations between users' acoustic properties of listening behavior and socio-economic features (WHR). Correlations >0.1 are highlighted in bold face. Statistically significant correlations at $p < 0.001$ are marked with an asterisk (*). (source: reference [6])

Dataset	**Items**	**Users**	**Ratings**	**Density (%)**	**User features**	**Items features**
MovieLens 100k	1,682	943	100,000	6.30	Age, gender, and occupation	Genres and year
Haodf	12,000	58,000	220,000	4.10	Doctors' positional titles	State of an illness

Table 11: Statistics of the two datasets used in this paper. (source: reference [5])

#Requests	1,480
#Request Authors	1,244
#Movies in Requests	5,521
#Unique Movies in Requests	1,908
#Requests With Positive Movies	1,480
#Requests With Negative Movies	77
#Keywords in Requests (Without Common Words)	4,492 (3,947)
#Unique Keywords in Requests (Without Common Words)	1,878 (1,762)
#Requests With Positive Keywords	1,202
#Requests With Negative Keywords	152
#Genres in Requests	762
#Unique Genres in Requests	25
#Requests With Positive Genres	459
#Requests With Negative Genres	55
#Actors in Requests	100
#Unique Actors in Requests	79
#Requests With Positive Actors	73
#Requests With Negative Actors	7
#Suggestions	43,402
#Unique Suggestions	6,071
#Suggestion Authors	7,431
Average #Suggestions per Request	29.33

Average Duration Between Request and Suggestion	31 h 41 min

Table 12: Reddit dataset statistics. (source: reference [7])

Model \ Metric	Precision	Recall	F1 score
Keywords			
BiLSTM (ELMO)	0.514 [0.467 – 0.561]	0.498 [0.442 – 0.552]	0.506 [0.465 – 0.548]
RoBERTa large	0.502 [0.454 – 0.550]	0.413 [0.359 – 0.466]	0.454 [0.411 – 0.498]
BERT base multilingual cased	0.598 [0.536 – 0.661]	0.301 [0.255 – 0.348]	0.400 [0.352 – 0.450]
BERT large cased	0.598 [0.542 – 0.654]	0.405 [0.353 – 0.455]	0.483 [0.436 – 0.531]
Genres			
BiLSTM (ELMO)	0.775 [0.699 – 0.858]	0.723 [0.647 – 0.805]	0.748 [0.690 – 0.816]
RoBERTa large	0.735 [0.590 – 0.885]	0.181 [0.109 – 0.248]	0.291 [0.196 – 0.387]
BERT base multilingual cased	0.763 [0.686 – 0.847]	0.613 [0.532 – 0.698]	0.680 [0.616 – 0.753]
BERT large cased	0.797 [0.724 – 0.875]	0.662 [0.583 – 0.745]	0.723 [0.663 – 0.793]
Actors			
BiLSTM (ELMO)	0.156 [0.000 – 0.282]	0.294 [0.017 – 0.526]	0.204 [0.017 – 0.367]
RoBERTa large	0.000 [0.000 – 0.000]	0.000 [0.000 – 0.000]	0.000 [0.000 – 0.000]
BERT base multilingual cased	0.353 [0.063 – 0.581]	0.400 [0.133 – 0.646]	0.375 [0.135 – 0.607]
BERT large cased	0.279 [0.010 – 0.465]	0.333 [0.128 – 0.524]	0.303 [0.106 – 0.488]
Movies			
BiLSTM (ELMO)	0.798 [0.754 – 0.843]	0.673 [0.623 – 0.721]	0.730 [0.692 – 0.768]
RoBERTa large	0.773 [0.728 – 0.818]	0.476 [0.415 – 0.536]	0.589 [0.539 – 0.641]
BERT base multilingual cased	0.716 [0.675 – 0.760]	0.438 [0.398 – 0.479]	0.544 [0.507 – 0.582]
BERT large cased	0.700 [0.650 – 0.750]	0.428 [0.390 – 0.465]	0.531 [0.496 – 0.567]

Table 13: Results Associated With Positive Entities. (source: reference [7])

3.2. Data Cleaning and Preprocessing

Cleaning and preparing data play a vital role in setting up the groundwork for a music recommendation system. It is crucial to process the information collected from different sources, including user data, music data, and user-system interaction behavior data, to ensure that the algorithm receives clean and efficient data for better system performance.

Throughout the data collection and transmission phases, challenges like missing data, data noise, and data duplication may surface. These challenges can have a negative impact on subsequent model training and lead to decreased model robustness. To tackle these issues, systematic processing techniques are implemented to clean and convert the data.

A key aspect of data processing involves feature selection. Given the multitude of attributes in the dataset, it is essential to pinpoint the most pertinent features for processing to bolster the model's robustness. By focusing solely on the most relevant features, unnecessary information is sifted out, resulting in a more efficient recommendation system.

Moreover, it is not sufficient to just ensure recommendation accuracy; diversity in recommendation outcomes is also crucial. As user preferences evolve and online services expand, offering a range of recommendations based on personalization and precision becomes indispensable. For instance, if a user frequently listens to ancient songs, the recommendation list should encompass some variety such as melancholic tunes that differ from ancient styles.

The success of recommendations relies heavily on users' distinct historical behavioral data. By drawing insights from past behaviors and updating recommendations in real-time based on short-term and long-term inputs, the system can deliver customized experiences to users. This continuous learning process enhances personalization and precision as time progresses.

In summary, meticulous data cleaning and preprocessing are vital for optimizing the performance of a music recommendation system. By tackling issues like missing data and noise while selecting pertinent features for processing, the system can furnish accurate and diverse recommendations tailored to individual user preferences. See references: [3], [10].

Notation	Description
N	Number of patients
M	Number of doctors
K	Dimension of latent factors
D	Dimension of sequential features
$R \in R^{N \times M}$	Rating matrix
$U \in R^{N \times K}$	Latent factors of patients
$V \in R^{M \times K}$	Latent factors of doctors
$X \in R^{N \times D}$	Sequential features of patients
$Y \in R^{M \times D}$	Sequential features of doctors

Table 14: Summary of notations. (source: reference [5])

Symbol	Description
u	User
v	Item
M	Number of users
N	Number of items
G	Knowledge graph
(h,r,t)	Three-tuple

Symbol	Description
h, r, t	Head entity, tail entity, relationship
θ	A parameter of the function f
Y	A user-item interaction matrix
W,$\text{W}_{\text{l}}^{\text{VV}}$,$\text{W}_{\text{l}}^{\text{EV}}$,$\text{W}_{\text{l}}^{\text{VE}}$,$\text{W}_{\text{l}}^{\text{EE}}$	Weight
b,$\text{b}_{\text{l}}^{\text{V}}$,$\text{b}_{\text{l}}^{\text{E}}$	Bias
S(v),S(u)	A set of entities associated with music item item v (user u)
$f_{R\text{S}}$	The prediction probability of the recommendation module
$\alpha ()$	The tanh function
$\sigma ()$	The Sigmoid function
$\beta ()$	The softsign function
$\delta ()$	The softplus function
f_{KG}	The probability prediction of the KGE module
d	The dimensions of the hidden layer
C_{l}	The cross feature matrix of layer l
$\text{v}_{\text{l}+1}$,$\text{e}_{\text{l}+1}$	Item and entity feature vectors for layer l + 1 after cross-compression unit processing
λ1,λ2	The equilibrium parameters
J	The cross-entropy function

Table 15: Symbol description. (source: reference [1])

Experiment name	RMSE for:	Q1	Q2	Q3
Item-based with ratings similarity ($k=50$)	1.192	1.173	1.052	
User-based with ratings similarity ($k=10$)	1.326	1.313	1.199	
Item-based with all features similarity ($k=10$)	1.163	1.144	1.029	
User-based with all personalities similarity ($k=10$)	1.365	1.361	1.262	
Hybrid with ratings similarity ($k_s=50, k_u=10$)	1.180	1.163	1.043	
Hybrid with all personalities and features similarity ($k_s=10, k_u=10$)	1.150	1.139	1.028	

Table 16: RMSE (10-CV) calculated in baseline experiments, for each rating type. The k denotes the number of neighbors used for prediction, chosen experimentally (source: reference [24])

4. Text Preprocessing

4.1. Tokenization

Segmentation is a pivotal process in the text preparation phase for constructing a Music Recommendation System. It entails dividing the text into distinct segments, typically words or phrases. When dealing with music data, segmentation aids in converting the lyrics or song titles into a format compatible with the recommendation algorithm. By segmenting the text, we can scrutinize and extract meaningful insights from the lyrics, such as identifying crucial words or phrases that characterize the song's style or genre.
Segmentation is critical for feature extraction in the recommendation system. Once the text has been segmented, additional processing steps like eliminating stopwords and performing lemmatization can be implemented to enhance the quality of the extracted features. In this manner, segmentation forms the basis for converting raw text data into a structured format suitable for developing a content-based recommender model.
In essence, segmentation plays a pivotal role in transforming textual content from songs into organized information that can be leveraged by the recommendation system to produce personalized music suggestions based on user preferences and song attributes. See references: [15], [14].

Model \ Metric	**P**	**R**	**F1**
Base BiLSTM 1 (dropout 0.1, units 100, recurrent dropout 0.3)	0.42	0.33	0.34
Base BiLSTM 2 (dropout 0.5, units 100, recurrent dropout 0.3)	0.47	0.30	0.32
Base BiLSTM 3 (dropout 0.3, units 150, recurrent dropout 0.3)	0.42	0.30	0.32
Base BiLSTM 4 (dropout 0.5, units 200, recurrent dropout 0.7)	0.37	0.34	0.33
Base BiLSTM 5 (dropout 0.5, units 150, recurrent dropout 0.7)	0.41	0.32	0.33
Features (143) 1 (dropout 0.3, units 200, recurrent dropout 0.5)	0.39	0.29	0.31
Features (143) 2 (dropout 0.3, units 150, recurrent dropout 0.5)	0.40	0.33	0.35

Model \ Metric	P	R	F1
Features (143) 3 (dropout 0.1, units 100, recurrent dropout 0.3)	0.43	0.32	0.35
Features (40) 1 (dropout 0.1, units 200, recurrent dropout 0.4)	0.41	0.35	0.36
Features (40) 2 (dropout 0.3, units 100, recurrent dropout 0.5)	0.39	0.33	0.36
Features (50) 1 (dropout 0,1, units 100, recurrent dropout 0.5)	0.46	0.32	0.34
Features (50) 2 (dropout 0.1, units 100, recurrent dropout 0.3)	0.47	0.33	0.34
Features (50) 3 (dropout 0.3, units 100, recurrent dropout 0.5)	0.42	0.31	0.32
ELMO 1 (dropout 0.2, units 512, recurrent dropout 0.35)	0.52	0.39	0.43
ELMO 2 (dropout 0.5, units 512, recurrent dropout 0.35)	**0.54**	**0.45**	**0.48**
ELMO 3 (dropout 0.35, units 150, recurrent dropout 0.35)	0.50	0.40	0.42
ELMO 4 (dropout 0.35, units 512, recurrent dropout 0.5)	0.51	0.43	0.46
ELMO features (40) 1 (dropout 0.3, units 512, recurrent dropout 0.5)	0.53	0.46	0.49
ELMO features (40) 2 (dropout 0.2, units 512, recurrent dropout 0.5)	0.51	0.43	0.45
ELMO features (40) 3 (dropout 0.2, units 512, recurrent dropout 0.3)	0.53	0.42	0.45
ELMO features (50) 1 (dropout 0.3, units 512, recurrent dropout 0.7)	0.51	0.42	0.45
ELMO features (50) 2 (dropout 0.1, units 512, recurrent dropout 0.7)	0.52	0.40	0.43
ELMO features (50) 3 (dropout 0.7, units 512,	0.52	0.43	0.45

Model \ Metric	P	R	F1
recurrent dropout 0.5)			

Table 17: Feature-Based Models Average Results. (source: reference [7])

Parameter name	Parameter setting	Description
CNN layer	2	The number of CNN layers
HiddenSize	16	The number of hidden layers
Filters	2	The number of filters
KernelSize	4	The number of kernels
Strides	2	The number of strides
Activation	ReLU	Activation function
Max pooling	1	The number of max poolings
Flattened	1	Flattened convolution
Fully connected	1	Fully connected layer
Dropout%	0.10	Discard rate
λ	0.01	Regularization coefficient

Table 18: PMF-CNN baseline architecture. (source: reference [5])

4.2. Stopword Removal

Eliminating filler words is a key component of text preprocessing when developing a music recommendation system. These filler words, known as stopwords, such as 'and', 'the', or 'is', do not contribute significantly to the text's meaning. By getting rid of these stopwords, we can focus on the important words that provide valuable information for our recommendation model.
When it comes to generating recommendations based on user-provided text, research and experiments involve a dataset that includes positively and negatively reviewed movies, keywords, genres, and actors. This dataset is crucial for understanding user preferences and creating precise recommendations. Removing stopwords from this dataset is essential for extracting relevant keywords and genres that are instrumental in building a strong content-based recommender system.
The use of automatic annotations yields scores for various annotation types like keywords, genres, actors, and movies. By eliminating stopwords from these annotations, we can enhance the accuracy of our recommendation model by honing in on the key terms that impact users' decisions.
During token-level evaluation of models like BERT and RoBERTa, precision, recall, and F1 scores are analyzed to fine-tune performance. Removing stopwords can boost the effectiveness of these models by removing extraneous words and emphasizing significant features that lead to improved recommendations.
In chi-square feature selection, which measures the relationship between terms and categories

in content-based recommendation systems, removing stopwords beforehand can amplify its impact by focusing on meaningful terms that shape relevant item recommendations.
Understanding how content is represented is vital for extracting features in recommendation systems. By removing stopwords from textual content and other elements like image recognition or collaborative filtering data, we can extract valuable insights that serve as the foundation for our recommendation algorithms.
Data labeling is crucial for sentiment analysis of songs sourced from Twitter. Removing stopwords from labeled texts enhances sentiment classification accuracy by highlighting pertinent words that convey positive or negative emotions linked to songs.
In the realm of online doctor recommendations utilizing CNN with sparse inputs, TF-IDF feature extraction is used to extract crucial features from patient reviews. By eliminating stopwords before calculating TF-IDF values, we prioritize important terms in reviews for more accurate doctor recommendations.
Ultimately, eliminating stopwords is a critical step in building a music recommendation system as it facilitates extracting essential information from text data and refining the accuracy of content-based filtering algorithms. See references: [7], [4], [8], [5].

Model \ Metric	P	R	F1
BERT base (uncased)	0.40	0.40	0.37
BERT base (cased)	0.50	0.39	0.42
BERT large (uncased)	0.41	0.29	0.33
BERT large (cased)	**0.55**	**0.40**	**0.45**
BERT base multilingual (uncased)	0.44	0.40	0.39
BERT base multilingual (cased)	**0.56**	**0.38**	**0.43**
RoBERTa base	0.36	0.23	0.27
RoBERTa large	**0.58**	**0.37**	**0.42**
XLM RoBERTa base	0.13	0.07	0.06

Table 19: Fine-Tuned Models Average Results. (source: reference [7])

A=#(t,c)	C=#(¬t,c)
B=#(t,¬c)	D=#(¬t,¬c)
N=A+B+C+D	

Table 20: Two-way contingency table for 2. (source: reference [8])

	Text	Spotify Key	Vector	Polarity
1	Estoy obsesionado	48PBbjlk x7n3cICgAVJXR M	[0,7712, -0.123, …, 1.2343]	N

2	Te extraño	2tFwfmc eQa1Y6nRPhYb EtC	[0.0198 , 0.125, …, -0.7322]	N
… .	….	…	…	…
411 4	No arrepiento nada	1LOuZm emeDN1SkJc9Ee mFI	[0.0128 , 0.156, …, -0.2701]	P

Table 21: Labeled Twitter dataset (source: reference [4])

4.3. Lemmatization

Normalization is a fundamental aspect of text preprocessing that involves reducing words to their base or root forms. This process aids in standardizing textual data by eliminating inflections and variations, thereby enhancing the efficiency of the recommendation system. Particularly in the realm of music recommendation systems, normalization plays a pivotal role in accurately identifying similar songs based on their content.
By implementing normalization on textual data, we can transform words into their base forms, facilitating the identification of common themes or genres present in music tracks. This approach enriches the feature extraction stage by simplifying the analysis of audio content characteristics and similarity metrics utilized in content-based music recommendation systems.
Normalization contributes to establishing a more organized and cohesive dataset for feature extraction procedures, such as TF-IDF vectorization. By standardizing words to their base forms, it enables better comparison and recognition of patterns within the music data. Consequently, this aids in constructing a resilient recommender model founded on precise and relevant features extracted from the normalized text.
To summarize, normalization stands as a crucial preprocessing method that significantly influences the caliber and precision of music recommendations produced by the system. It harmonizes textual data, enhances feature extraction, and boosts the overall performance of content-based filtering algorithms within music recommendation systems. See references: [9] p. 1-5, [10].

5. Feature Extraction

5.1. TF-IDF Vectorization

Feature extraction using TF-IDF vectorization is a prevalent technique employed in music recommendation systems to derive key attributes from the items within the system. This strategy entails constructing a user profile based on a weighted vector of item characteristics in a content-based manner. The significance of each feature is indicated by the assigned weights, which can be computed through diverse methodologies. These can range from simplistic methods like averaging values of rated item vectors to more sophisticated approaches involving machine learning algorithms such as Bayesian classifiers, cluster analysis, decision trees, and artificial neural networks.

A critical aspect of content-based filtering is the system's capability to learn user preferences from one genre of content and extrapolate them to other genres. This functionality is vital for augmenting the utility of the recommendation system, as it enables suggestions from various platforms beyond the user's current usage. Moreover, opinion-driven recommender systems are integrated into content-based frameworks, allowing users to furnish text reviews or feedback on items. These user-generated texts serve as implicit data for the recommendation system and offer valuable insights into both item characteristics and user sentiments.
In the realm of music recommendation systems, TF-IDF is synergistically employed with deep convolutional neural networks (DCNN) and weighted feature extraction (WFE) methods to establish correlations between user data and music data. The TF-IDF strategy generates implicit user ratings for songs based on term frequency and inverse document frequency computations. These implicit ratings are subsequently utilized by the EPMRS (Explicit Preference Music Recommendation System) to suggest music tracks to users based on their preferences. Empirical findings have demonstrated that EPMRS surpasses baseline systems like Content Similarity Music Recommendation System (CSMRS) and Personalized Music Recommendation System based on Electroencephalography feedback (PMRSE) in terms of accuracy.
In conclusion, TF-IDF vectorization plays a pivotal role in extracting significant features for tailored music recommendations, thereby enhancing user satisfaction and overall experience with music streaming services. See references: [13], [17], [5].

6. Building a Recommender Model & Content-Based Filtering

6.1. Similarity Measures (Cosine Similarity)

Measuring similarity is a fundamental aspect of music recommendation systems, as it allows for assessing the closeness between different music pieces. One widely used method for measuring similarity in these systems is cosine similarity, which calculates the cosine of the angle between vectors representing music items. The smaller the angle, the more similar the items are considered to be.
In the realm of music recommendation algorithms that rely on knowledge graphs and multi-task feature learning, cosine similarity is a valuable tool for evaluating how closely related music items are based on their extracted features. By exploring various functions like sigmoid, tanh, softsign, and softplus to compute the true and false triplet scores of music triplets, cosine similarity aids in identifying similarities between music items within the recommendation system. Furthermore, cosine similarity contributes to improving recommendation accuracy by uncovering patterns and subtle differences that traditional methods may miss. By harnessing AI technologies such as machine learning and deep learning, cosine similarity enhances the ability to predict user preferences accurately and deliver personalized recommendations tailored to individual needs.
In summary, cosine similarity plays a significant role in constructing effective music recommendation systems by quantifying similarities between music items based on their features from knowledge graphs and facilitating the generation of personalized

recommendations for users. See references: [1], [2].

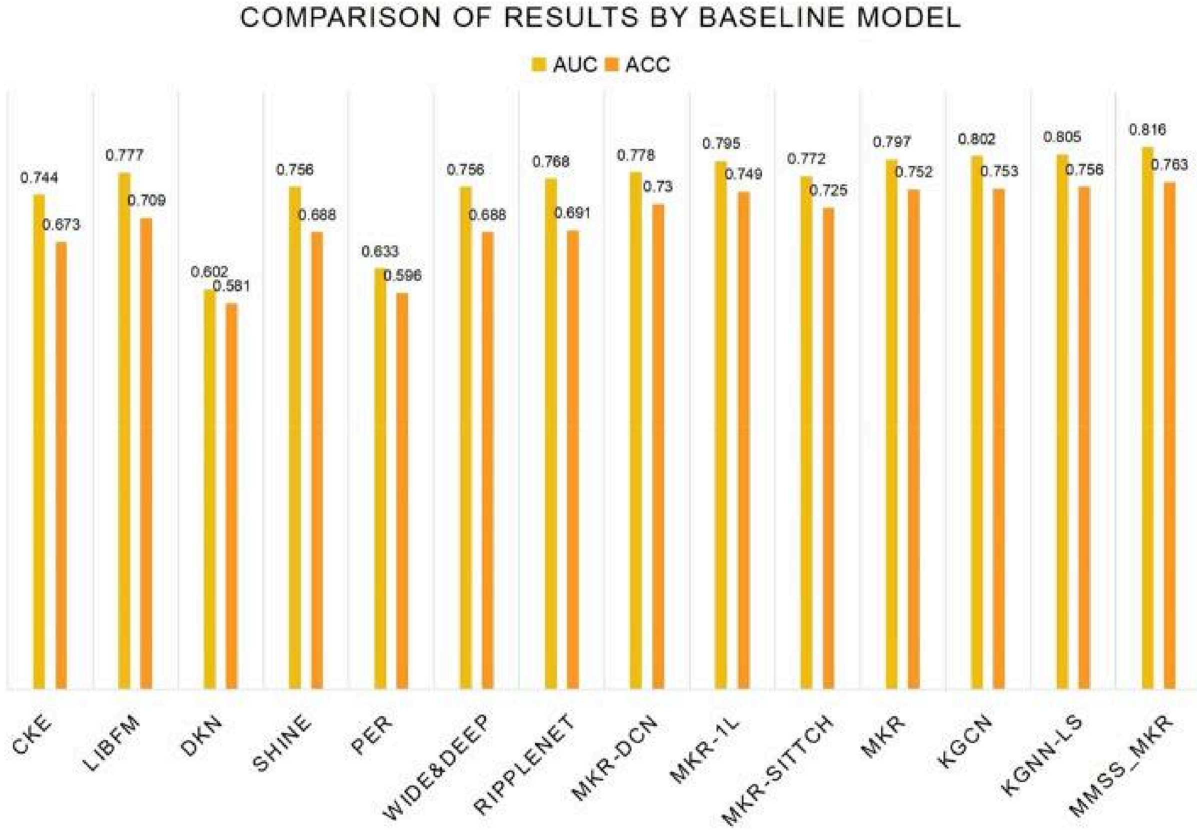

Figure 3: Histogram comparing results across baseline models. (source: reference [2])

Model	Last.FM	
AUC	ACC	
CKE	0.744	0.673
LibFM	0.777	0.709
DKN	0.602	0.581
SHINE	0.756	0.688
PER	0.633	0.596
Wide&Deep	0.756	0.688
RippleNet	0.768	0.691
MKR-DCN	0.778	0.730
MKR-1L	0.795	0.749
MKR-sittch	0.772	0.725
MKR	0.797	0.752
KGCN	0.802	0.753

Model	Last.FM	
AUC	**ACC**	
KGNN-LS	0.805	0.756
MMSS_MKR	**0.816**	**0.763**

Table 22: Comparison of modeling results for music datasets. (source: reference [2])

6.2. Building a Content-Based Recommender System

Developing a content-based recommender system involves harnessing the power of content-based filtering to offer precise and personalized suggestions to users. By examining the content of music items to deduce user preferences and propose similar items based on extracted features, the system can cater to both popular and niche music, effectively tackling the issue of new user initialization.

One strategy for creating a content-based recommender system is through deep learning models that automate feature extraction from audio content and provide tailored recommendations. These models combine feature learning with recommendation stages, surpassing traditional methods by capturing more pertinent information in audio data. Moreover, integrating these acquired features with collaborative filtering in a hybrid approach further boosts recommendation performance without solely depending on CF.

Examples such as Spotify's music recommendation algorithm showcase the adaptability and efficiency of content-based recommendation systems across various fields. By utilizing content characteristics, user actions, and evolving profiles, these systems offer customized experiences that align with individual preferences. As technology progresses, content-based recommendation systems will continue to be crucial in personalized content discovery, shaping the landscape of personalized content distribution. See references: [22], [18] p. 11-15, [20].

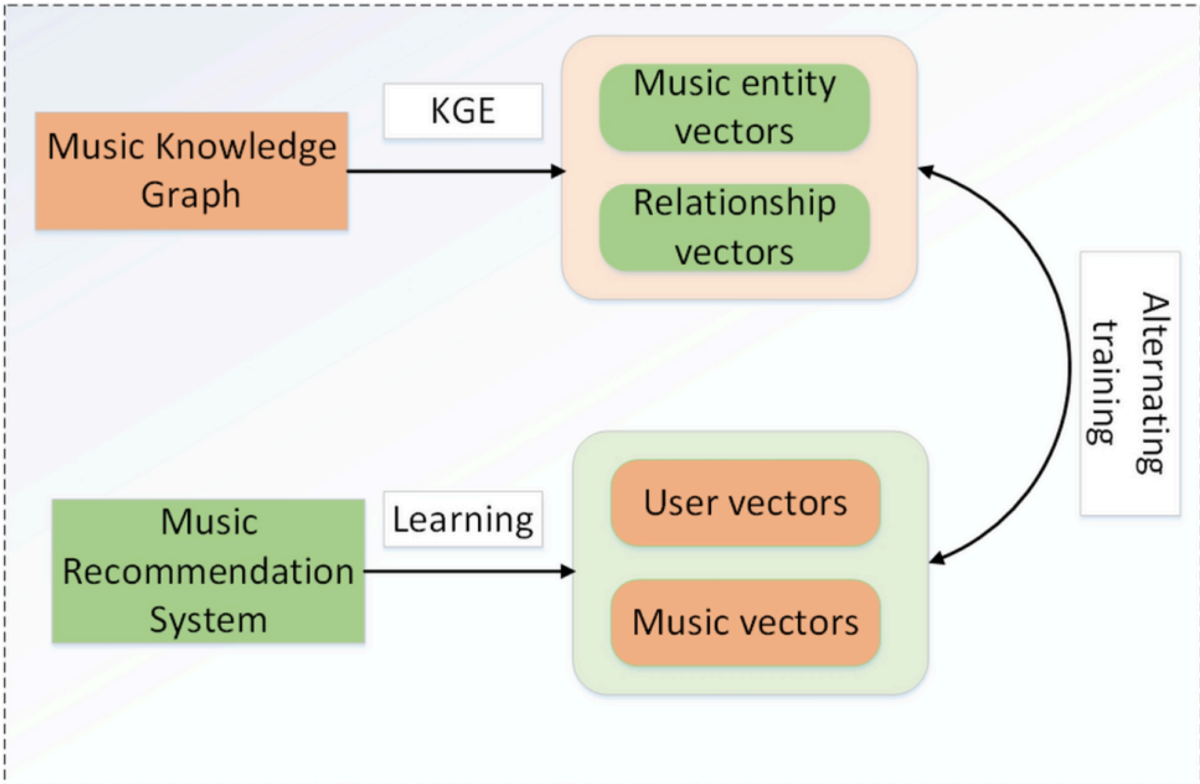

Figure 4: Structure of alternate training of models. (source: reference [1])

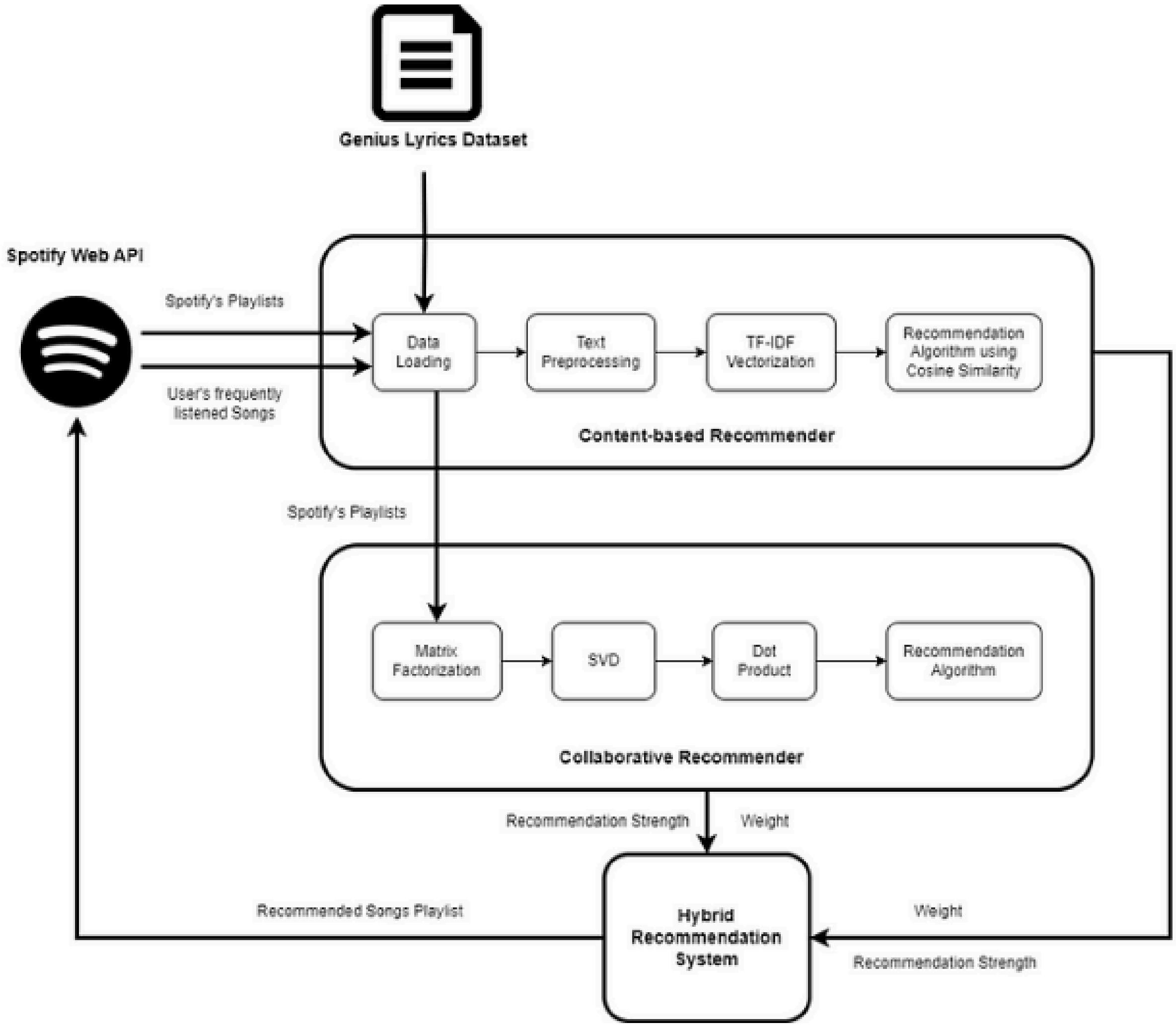

Figure 5: System Architecture (source: reference [25])

7. User Interaction and Recommendations

7.1. Designing User Interface for Recommendations

Creating a user-friendly interface for music recommendations plays a crucial role in keeping users engaged and satisfied. The interface should strive to offer a smooth and easy-to-navigate platform for users to explore new music that resonates with their tastes. Drawing insights from various data sources, the interface can be tailored to showcase personalized recommendation lists based on individual preferences.
An essential aspect to consider when designing the user interface is finding the right balance

between novelty and familiarity, diversity and similarity, as well as popularity and personalization. By presenting a mix of beloved tracks alongside suggestions for new discoveries, the interface can accommodate different user tastes and enrich the overall listening experience. Additionally, incorporating features that allow users to provide feedback on recommendations can help refine the system over time and enhance the accuracy of future suggestions.

Transparency also plays a crucial role in user interface design. Users should have visibility into how the recommendation system operates and why specific songs are recommended to them. By offering explanations for recommendations based on content features or descriptions, users can develop trust in the system and feel more confident exploring new music suggestions. Moreover, real-time monitoring of user interactions can empower the system to update personalized recommendation lists dynamically based on user behavior. This continuous feedback loop ensures that recommendations remain relevant and tailored to each individual's evolving preferences.

In conclusion, crafting a user interface for music recommendations involves creating a seamless experience that strikes a balance between familiarity and novelty, incorporates transparency in recommendation explanations, and leverages real-time feedback to deliver personalized suggestions. By focusing on these elements in interface design, users can enjoy a more immersive and satisfying music discovery journey. See references: [13], [18] p. 6-10, [10].

```
recommend_songs('understand')
```

	Name	Artist
2	Bandana	Fireboy DML
32	Rush	Ayra Starr
15	Cold Outside	Timaya
48	emiliana	CKay
41	Bad To Me	Wizkid
35	B. D'OR (feat. Wizkid)	Burna Boy
9	Buga (Lo Lo Lo)	Kizz Daniel

Figure 6: It recommended 7 songs (source: reference [12])

7.2. Providing Real-Time Recommendations

In today's fast-paced digital landscape, real-time recommendation systems have become indispensable. Users now demand immediate and tailored suggestions, which these systems can provide by adjusting recommendations on the fly during a user session. Thanks to sophisticated software algorithms, real-time recommendation systems can analyze user behavior in real time to offer personalized suggestions. This is a stark contrast to traditional batch recommendation systems that rely on static data sets and lengthy ETL workflows, as

real-time systems can swiftly adapt based on new user interactions.
The integration of machine learning is paramount in powering real-time recommendation systems, enabling them to anticipate user preferences and deliver personalized suggestions with precision. Through the utilization of online machine learning models and advanced data processing techniques, these systems can continuously optimize the model based on user feedback. This adaptive approach ensures that the system can effectively handle dynamic and unpredictable user behavior, guaranteeing that recommendations are always relevant and timely.
Content-based filtering is a prevalent method used in real-time recommendation systems to analyze an individual user's past actions, such as previously watched movies or purchased items. By identifying content similar to what the user has enjoyed before, these recommenders can suggest relevant content tailored to individual tastes. The hybrid approach combines content similarity with weighted popularity to offer highly personalized recommendations across various music genres or categories.
Ultimately, real-time recommendation systems driven by machine learning models are instrumental in enhancing user engagement, boosting customer satisfaction, increasing sales and subscription rates, and driving revenue growth across different online platforms. By harnessing intelligent algorithms and adapting dynamically to user interactions, these systems provide valuable insights into user behavior and enable data-driven marketing decisions. See references: [21], [22], [20].

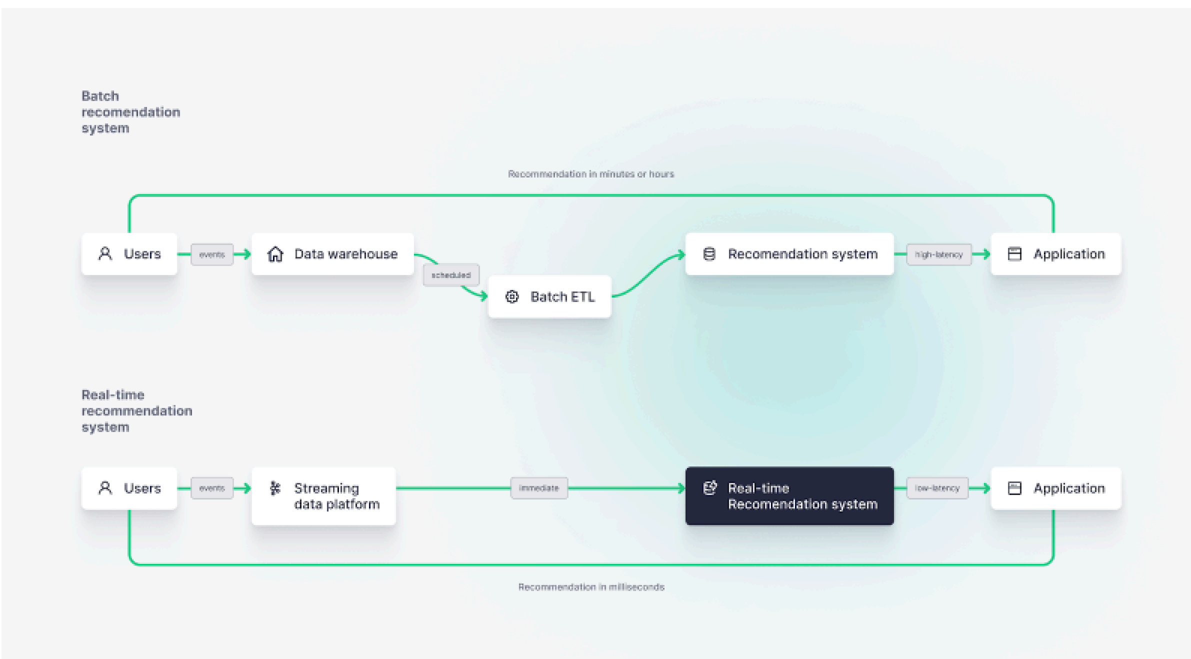

Figure 7: Unlike batch recommendation systems, real-time recommendation systems use streaming data platforms and real-time databases to make recommendations to users as they are browsing and based on the most up-to-date data. (source: reference [21])

8. Evaluation of the Recommendation

System Performance

In assessing the effectiveness of the Music Recommendation System, a series of metrics and experiments have been carried out using data obtained from various sources. The MMSS_MKR model, designed to enhance music recommendations by leveraging knowledge graphs and multi-task feature learning, has demonstrated significant advancements in comparison to existing models. The evaluation metric Area Under the Curve (AUC) achieved a score of 0.816, while Accuracy (ACC) reached 0.763 when evaluated on the Last.FM music dataset. These outcomes underscore the value of integrating knowledge graph information into recommendation systems to boost recommendation accuracy.

Furthermore, experiments involving hyperparameters were conducted to analyze different configurations and their impact on performance metrics like AUC and ACC. By adjusting parameters such as L and H, enhancements in recommendation accuracy were observed, with higher values leading to superior outcomes. This highlights the importance of fine-tuning hyperparameters to optimize the performance of recommendation systems.

Moreover, user modeling strategies based on cultural cues and acoustic features were assessed to evaluate their influence on recommendation quality. Models that combined music and cultural characteristics exhibited increased precision, recall, and F1-score compared to models focusing solely on music or geographical information. This suggests that integrating cultural cues into user modeling can elevate recommendation accuracy and relevance.

In summary, the evaluation of the Music Recommendation System's performance underscores the effectiveness of incorporating knowledge graphs, fine-tuning hyperparameters, and integrating cultural cues into user modeling. These strategies aim to enhance recommendation accuracy and provide users with more personalized suggestions. See references: [1], [2].

Hyperparameter	**Last.FM**	
AUC	**ACC**	
L = 1,H = 1	0.801	0.751
L = 1,H = 2	0.809	0.756
L = 1,H = 3	0.812	0.759
L = 1,H = 4	0.816	0.763
L = 1,H = 5	0.813	0.761

Table 23: Hyperparameter analysis. (source: reference [1])

Model	**User Features**	**Track Features**
Music + Culture	U_ID, U_AF, U_WHR, U_HOF	T_ID, T_AF, T_WHR, T_HOF
Music	U_ID, U_AF	T_ID, T_AF
Culture	U_ID, U_WHR, U_HOF	T_ID, T_HOF, T_WHR

Model	User Features	Track Features
Country	U_ID, U_Country_ID	T_ID, T_Country_ID
User + Track	U_ID	T_ID

Table 24: Overview of evaluated models, where features prefixed with U describe a user and features prefixed with T describe a track; the models on two last rows serve as baselines. (source: reference [6])

Model	Prec	Rec	F_1
Music + Culture	0.98 (±0.04)	0.63 (±0.15)	0.75 (±0.10)
Music	0.95 (±0.06)	0.59 (±0.15)	0.72 (±0.11)
Country	0.83 (±0.11)	0.52 (±0.12)	0.63 (±0.10)
Culture	0.31 (±0.15)	0.18 (±0.08)	0.24 (±0.09)
User + Track	0.13 (±0.10)	0.08 (±0.06)	0.13 (±0.06)

Table 25: Precision, recall, and F 1 -score for all proposed models (sorted by performance; standard deviation in parentheses). (source: reference [6])

Model	RMSE	MAE
Music + Culture	0.15	0.02
Music	0.17	0.03
Country	0.36	0.13
Culture	0.88	0.77
User + Track	0.93	0.85

Table 26: RMSE and MAE of all models. (source: reference [6])

Entities	Sentiment		[CI]		[CI]
Movies	all pos.	0.1174	[0.1033,0.1309]	0.1148	[0.1008,0.1283]
pos./neg.	0.1185	[0.1044,0.1320]	0.1148	[0.1008,0.1283]	
Movies | Genres	all pos.	0.1201	[0.1059,0.1337]	0.1173	[0.1029,0.1310]
pos./neg.	0.1204	[0.1061,0.1340]	0.1168	[0.1024,0.1306]	
Movies | Keywords	all pos.	0.1250	[0.1107,0.1387]	0.1211	[0.1068,0.1349]
pos./neg.	0.1253	[0.1112,0.1390]	0.1228	[0.1085,0.1365]	
Movies | Actors	all pos.	0.1152	[0.1016,0.1284]	0.1145	[0.1004,0.1280]
pos./neg.	0.1183	[0.1044,0.1316]	0.1145	[0.1004,0.1280]	
Movies | Genres | Keywords | Actors	all pos.	0.1244	[0.1103,0.1380]	0.1228	[0.1083,0.1368]
pos./neg.	**0.1257**	[0.1115,0.1395]	**0.1244**	[0.1099,0.1384]	

Table 27: Recommender results. (source: reference [7])

Dataset	Haperparameters
Last.FM	L = 1 H = 1 dim = 128 n_epochs = 200 Batch_size = 256 Lr_rs = 2×10−5 Lr_kge = 2×10−5 l2_weight = 2×10−5 kge_interval = 64

Table 28: Hyperparameter settings. (source: reference [2])

9. Conclusion and Future Work

In summary, the creation of a music recommendation system utilizing Python and machine learning presents a vast opportunity to deliver personalized suggestions to users. By making use of the Spotify dataset, we can gather music information, preprocess it, and extract features using methods like TF-IDF vectorization. Employing content-based filtering and similarity metrics such as cosine similarity enables us to construct a strong recommender model capable of providing precise recommendations.
Looking ahead, it is crucial to concentrate on user engagement and real-time suggestions by designing a user-friendly interface and ensuring smooth integration for delivering timely recommendations. Additionally, assessing the performance of the recommendation system is essential for gauging its efficacy and implementing necessary enhancements.
Future endeavors in this area may include enhancing dataset quality by taking into account larger datasets and exploring new features like lyrics analysis through Natural Language Processing techniques. Integrating various recommendation approaches, such as

content-based and collaborative filtering methods, could further bolster the system's accuracy. Furthermore, incorporating hybrid recommendation systems that merge different filtering techniques can help overcome individual method drawbacks.
On the whole, developing a music recommendation system is an engaging field that demands ongoing innovation to meet evolving user demands. By integrating advanced techniques and strategies, we can establish an effective and user-focused music recommendation system that elevates the overall listening experience. See references: [12], [19], [18] p. 66-70, [9] p. 6-10.

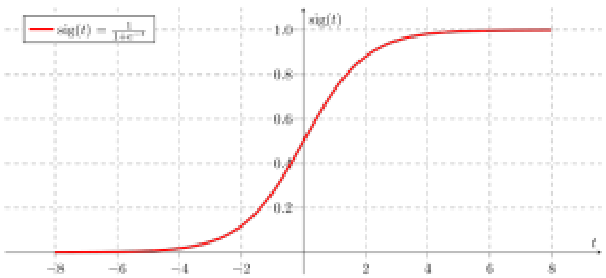

Figure 8: Sigmoid function (source: reference [18])

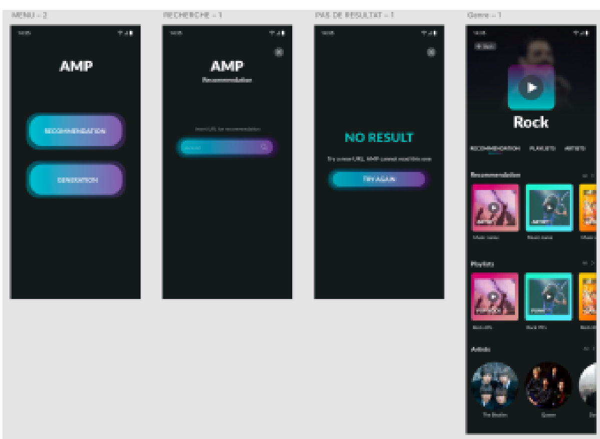

Figure 9: Main menu and recommendation part of the application (source: reference [18])

```
hybrid_model_recommendations_name[['id', 'name', 'artist_name', 'album_name', 'recStrengthHybrid']]
```

	id	name	artist_name	album_name	recStrengthHybrid
0	3AWVz5rK8Hrqo9YGiVGN5	Apocalypse	Cigarettes After Sex	Cigarettes After Sex	2.000000
1	4uzUuDjoulcBXYZ27CsLIF	Roses	Shawn Mendes	Illuminate (Deluxe)	0.148095
2	5lonbH68pmDKBwcKlyswmR	John Wayne	Cigarettes After Sex	Cigarettes After Sex	0.121117
3	07A4mFpXnuUrcRlc1S8llz	Bad Reputation	Shawn Mendes	Illuminate	0.106317
4	5ld8E3DVBXCCWQ54yrkWVp	305	Shawn Mendes	Wonder	0.075126
5	79esEXlqqmqOGPz0xQSZTV	Lost In Japan	Shawn Mendes	Shawn Mendes	0.069296
6	64GT5ReOko7xx6Nc7D8JsS	Honest	Shawn Mendes	Illuminate (Deluxe)	0.059194
7	0Qr61NXlyAeQaADO5xn3rl	Cry	Cigarettes After Sex	Cry	0.058108
8	1Bh0UzthW8pKEnYg7v40Oa	Heavenly	Cigarettes After Sex	Cry	0.055941
9	7nuQ1LgM6l5FLaJmlODoqj	Higher	Shawn Mendes	Wonder	0.053040
10	1RRZSm4akqNyMOsPUhw4cb	Because I Had You	Shawn Mendes	Shawn Mendes	0.049749
11	3QGsuHI8jO1Rx4JWLUh9jd	Treat You Better	Shawn Mendes	Illuminate	0.043710
12	138oQeapLds5leJrdRfeUd	Intro	Shawn Mendes	Wonder	0.036533

Figure 10: Hybrid-based Recommendations (source: reference [25])

References

- [1] X. Liu, Z. Yang and J. Cheng. "Music recommendation algorithms based on knowledge graph and multi-task feature learning". Jan 2024. [Online]. Available: https://www.nature.com/articles/s41598-024-52463-z
- [2] X. Liu, Z. Yang and J. Cheng. "Music recommendation algorithms based on knowledge graph and multi-task feature learning". Jan 2024. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10808181/
- [3] A. Abdul, J. Chen, H. Liao and S. Chang. "An Emotion-Aware Personalized Music Recommendation System Using a Convolutional Neural Networks Approach". Jul 2018. [Online]. Available: https://www.mdpi.com/2076-3417/8/7/1103
- [4] A. L. S. Medina, R. S. Leon and Hugo D. Calderon-Vilca. "Music Recommender System based on Sentiment Analysis Enhanced with Natural Language Processing Technics". Mar 2023. [Online]. Available: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-55462023000100053
- [5] Y. Yan, G. Yu and X. Yan. "Online Doctor Recommendation with Convolutional Neural Network and Sparse Inputs". May 2020. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7584959/
- [6] E. Zangerle, M. Pichl and M. Schedl. "User Models for Culture-Aware Music Recommendation: Fusing Acoustic and Cultural Cues". Mar 2020. [Online]. Available: https://transactions.ismir.net/articles/10.5334/tismir.37
- [7] L. Eberhard, K. Popova, S. Walk and D. Helic. "Computing recommendations from free-form text". Feb 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423017700
- [8] D. Wang, Y. Liang, D. Xu, X. Feng and R. Guan. "A content-based recommender system for computer science publications". Oct 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705118302107
- [9] J. Kaitila. "A content-based music recommender system". May 2017. [Online]. Available: https://core.ac.uk/download/pdf/250147513.pdf
- [10] P. Sun. "Music Individualization Recommendation System Based on Big Data Analysis". Jul 2022. [Online]. Available: https://www.hindawi.com/journals/cin/2022/7646000/
- [11] Y. Zhang. "Design of the Piano Score Recommendation Image Analysis System Based on the Big Data and Convolutional Neural Network". May 2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8642031/
- [12] E. (. Octavio). "Music Recommendation System Built With Python And Machine Learning". Jan 2023. [Online]. Available: https://levelup.gitconnected.com/music-recommendation-system-built-with-python-and-machine-learning-5c983fbc7c35
- [13] "Recommender system". Apr 2019. [Online]. Available: https://en.wikipedia.org/wiki/Recommender_system
- [14] N. V. Otten. "How To Build Content-Based Recommendation System Made Easy [Top 8 Algorithms & Python Tutorial]". Apr 2024. [Online]. Available: https://spotintelligence.com/2023/11/15/content-based-recommendation-system/
- [15] N. Saeed. "Music Recommendation System using machine learning | AI Driven

Songs System". May 2023. [Online]. Available: https://medium.com/@611noorsaeed/music-recommendation-system-using-machine-learning-ai-driven-songs-system-416857802eb6
- [16] A. Niyazov, E. Mikhailova and O. Egorova. "Content-based Music Recommendation System". May 2021. [Online]. Available: https://www.researchgate.net/publication/351863177_Content-based_Music_Recommendation_System
- [17] A. Abdul, J. Chen, H. Liao and S. Chang. "An Emotion-Aware Personalized Music Recommendation System Using a Convolutional Neural Networks Approach". Jul 2018. [Online]. Available: https://www.researchgate.net/publication/326280021_An_Emotion-Aware_Personalized_Music_Recommendation_System_Using_a_Convolutional_Neural_Networks_Approach
- [18] M. Chemeque-Rabel. "Content-based music recommendation system: A comparison of supervised Machine Learning models and music features". Aug 2020. [Online]. Available: http://www.diva-portal.org/smash/get/diva2:1515217/FULLTEXT01.pdf
- [19] N. Serhii Ripenko. "Music recommendation system: all you need to know". Nov 2022. [Online]. Available: https://www.eliftech.com/insights/all-you-need-to-know-about-a-music-recommendation-system-with-a-step-by-step-guide-to-creating-it/
- [20] A. Kharwal. "Music Recommendation System using Python". Jul 2023. [Online]. Available: https://thecleverprogrammer.com/2023/07/31/music-recommendation-system-using-python/
- [21] J. Karlsson. "What it takes to build a real-time recommendation system". (accessed May 09, 2024). [Online]. Available: https://www.tinybird.co/blog-posts/real-time-recommendation-system
- [22] X. Wang and Y. Wang. "Improving Content-based and Hybrid Music Recommendation using Deep Learning". Nov 2014. [Online]. Available: https://dl.acm.org/doi/10.1145/2647868.2654940
- [23] M. Auren, A. Baaw, T. Karlsson, L. Nilsson, D. H. Olzon and P. Shirmohammad. "Music Recommendations Based on Real-Time Data". May 2018. [Online]. Available: https://publications.lib.chalmers.se/records/fulltext/256144/256144.pdf
- [24] M. Klec, A. Wieczorkowska, K. Szklanny and W. Strus. "Beyond the Big Five personality traits for music recommendation systems". Jan 2023. [Online]. Available: https://asmp-eurasipjournals.springeropen.com/articles/10.1186/s13636-022-00269-0
- [25] A. Parab and S. Gajbhiye. "Personalized Music Playlist Recommendation System". Oct 2023. [Online]. Available: https://medium.com/@gajbhiyeshreya23/personalized-music-playlist-recommendation-system-776bfabd50c2