BACS3074

ARTIFICIAL INTELLIGENCE

202401 Session, Year 2023/24

Assignment Documentation

Project Title: Music Recommendation System
Programme: RDS Y2S2
Tutorial Group: GROUP 3
Tutor: DR NOOR AIDA BINTI HUSAINI

Team members' data

No	Student Name	Student ID	Module In Charge
1	Yam Jason	22WMR13662	GLOVE Algorithm
2.	Wong Yee En	22WMR13659	WORD2VEC Algorithm
3.	Ashantha Rosary James K Arokiasamy	22WMR14161	DOC2VEC Algorithm

Table of Contents

1. Introduction

1.1. Problem Background

Music lovers in today's digital world face a delightful but formidable issue. The goal is to find the perfect tracks amid a sea of limitless alternatives. Take Spotify for example; it's like having an entire world of music in your pocket. However, with millions of music to choose from, it might be difficult to discover something that truly speaks to you.

In the past, music recommendation systems typically used user ratings or playlists to recommend new songs. Nevertheless, these techniques are not always accurate in understanding the specific preferences and emotions of a listener. Input the requirement for additional personalised suggestions.

Spotify, as a major player in the music streaming industry, gathers a vast amount of data on our listening habits and interactions with songs. This wealth of data contains the answer to developing improved recommendation systems. However, the challenge lies in sorting through large amounts of data and providing recommendations that genuinely connect with each individual listener, which is no easy task (Feng et al., 2020).

Despite the advantages of methods such as collaborative filtering, they face challenges in situations known as cold start, which occur when there is limited user data available, such as when a user is new to the platform (Velankar & Kulkarni, 2023). This is where content-based recommendation systems become important.

Content-based systems thoroughly analyse features of each song such as lyrics, track title, and artist name (Kathavate, 2021). By comprehending the unique qualities of each track, these systems are able to provide personalised recommendations that cater to each listener's tastes.

But here's the rub, building a content-based recommendation system that really hits the mark is not easy. There are challenges around choosing the right features to analyse, building models that can handle the complexity of music data, and making sure recommendations strike the right balance between familiarity and surprise.

1.2. Objectives/Aims

Aim

- develop a robust song recommendation system capable of matching user predispositions with high accuracy and personalised recommendations. To achieve this overarching goal, several specific objectives have been outlined:

Objectives

1. Recommend songs based on user input by employing the GloVe algorithm, utilising metrics like Cosine Similarity, Jaccard Similarity, and Pearson Correlation Coefficient.
2. create a user-friendly Streamlit application translating recommendation algorithms, enabling users to input preferences and explore recommended songs for an enhanced user experience, maximising utility and impact.
3. Measure algorithm runtime to ensure computational efficiency for real-time recommendation generation in large-scale music libraries.

1.3. Motivation

The motivation behind this assignment lies in exploring the potential commercialization value and social impacts of innovative recommendation systems for the music industry.

Music streaming providers stand to gain substantial commercial prospects by improving user engagement and generating income through the development of sophisticated recommendation algorithms. These platforms can boost user happiness and retention, which can result in greater subscription rates and advertising income, by offering personalised recommendations that are catered to the tastes of individual users. Furthermore, through partnerships with record labels and artists, as well as targeted marketing and promotional efforts, these systems can uncover new revenue sources by utilising machine learning algorithms and rich music information.

Personalised recommendation systems have the potential to have significant social effects on how individuals find, listen to, and engage with music that go beyond the domain of commerce. These systems have the ability to create a sense of community by creating tailored playlists and recommendations that represent a range of musical preferences and cultural identities. Additionally, these platforms can help democratise the music industry by amplifying marginalised voices and establishing a more varied and inclusive musical environment by encouraging the discovery of up-and-coming musicians and niche genres.

In summary, the development of personalised recommendation systems for the music industry not only holds significant commercialization value in terms of driving user engagement and revenue generation but also has the potential to have far-reaching social impacts by promoting cultural diversity, inclusivity, and community engagement through music.

1.4. Timeline/Milestone

	Week										
	1	2	3	4	5	6	7	8	9	10	11
Project Research											
Project Planning											
Data Collection and Preprocessing											
Algorithm Implementation											
Testing and Optimization											
Integration											
Interface Developement											
Deployment and Evaluation											
Documentation											

2. Research Background

2.1. Background of the applications

The background of music recommendation systems (MRS) is rooted in the concept of filtering systems designed to predict user preferences for specific elements, such as songs. These systems have become an integral part of our daily routine, with examples including targeted advertising, personalised video playlists, and music recommendation engines like TikTok's. In the music industry, MRS are part of the core engine of streaming apps like Spotify, YouTube Music, Deezer, and Tidal, utilising machine learning and artificial intelligence to generate personalised playlists based on user listening history and preferences (Schedl, 2019).

Historically, research on MRS has emerged from two distinct communities, i.e., music information retrieval (MIR) and recommender systems (RS), with different focuses, perspectives, and terminologies. MIR has its origins in library science and signal processing and has focused strongly on content-based approaches, where "content" refers to information extracted from the actual audio signal (None Swarnima & Mala Saraswat, 2024). However, while much research in MIR has addressed the topic of audio similarity, which is a prerequisite to build content-based MRS, surprisingly little research by the MIR community has been devoted specifically to music recommendation.

Algorithms for recommendation systems (RS) can be broadly categorised into collaborative filtering (CF), context-aware (CA), content-based filtering (CBF), and hybrid approaches. Collaborative filtering models leverage historical user interactions or explicit preferences to identify patterns and predict item utility, with memory-based and model-based methods being common. Model-based approaches further divide into linear methods like matrix factorization and graph-based methods, as well as nonlinear methods such as deep neural networks. Context-aware models incorporate contextual information like time or location to enhance CF. Content-based filtering models use target user behavioural data and item content information, like text, image colours, or music rhythm, to make recommendations. Hybrid systems combine multiple approaches for improved recommendation accuracy, often employing fusion techniques (Deldjoo et al., 2020).

According to a journal by Sheikh Fathollahi and Razzazi (2021), they presented a music similarity measurement and recommendation system utilising convolutional neural networks (CNNs) to automatically curate user playlists. The system begins by classifying music genres using CNNs trained on three distinct databases. Subsequently, similarity measurement employs Euclidean distance and cosine similarity, with a focus on extracting features from Mel spectrograms in intermediate CNN layers. The recommendation system suggests songs based on the similarity of their feature vectors, with cosine similarity proving more effective than Euclidean distance, particularly when incorporating multiple audio features. The system achieves notable accuracy, up to 88.23%, in recommending similar songs by utilising cosine similarity on feature vectors extracted from CNN intermediate layers, especially with diverse audio feature inputs (Sheikh Fathollahi & Razzazi, 2021).

Utilising Convolutional Neural Networks (CNN), the music recommendation system discussed in Zhang's (2022) research article delivers personalised music suggestions by processing digital piano music and extracting comprehensive features from spectrum and notes. The system encompasses classification methods and recommendation algorithms, leveraging historical behaviour data and audio information to classify music based on user preferences and characteristics. Employing both content-based and collaborative filtering algorithms, it extracts audio data characteristics and analyses user behaviour to provide accurate suggestions. While offering personalised recommendations and improved accuracy compared to traditional methods, challenges such as the cold start problem with new users and increased data processing burden due to deep learning approaches are observed. Nonetheless, it represents an advanced fusion of technology and user preferences, leveraging CNN for efficient feature extraction and recommendation modelling (Zhang, 2022).

In their study, Mukhopadhyay et al. (2024) examine two distinct strategies, content-based filtering and K-Means clustering, within a music recommendation system framework, with the aim of aligning recommendations with user preferences. While content-based filtering suggests songs based on similarity to past user choices, K-Means clustering groups similar songs into clusters based on shared characteristics. Evaluation using the Silhouette index reveals high recommendation accuracy exceeding 99% for K-Means clustering, yet content-based filtering outperforms it. This research sheds light on refining personalised music recommendation systems, providing insights into methodological effectiveness. Content-based filtering tailors recommendations to user preferences and handles data sparsity effectively, while K-Means clustering identifies hidden relationships in music data but may lack accuracy compared to content-based filtering. By comparing these approaches on the Spotify dataset, the study offers valuable insights for optimising recommendation systems, enhancing user experiences in music selection (Mukhopadhyay et al., 2024).

The music recommendation system proposed by Wahyudi et al. (2020) is a content-based approach that suggests music to users based on genre, artist, and user ratings. Utilising one-hot encoding to represent content features, the system calculates user preferences and similarity between users to recommend songs with the highest ratings. It evaluates its performance through metrics like precision, recall, and accuracy, achieving an accuracy of 85% and a precision of 88.9%. While advantageous in recommending new music aligned with past preferences and suitable for scenarios with known item data but limited user information, drawbacks include reliance on potentially inaccurate user ratings and the risk of suggesting familiar or disliked songs (Wahyudi et al., 2020).

The music recommendation system described by Niyazov et al. (2021) operates on a content-based approach and also uses ANN, leveraging audio features like spectral centroid, spectral flatness, spectral roll-off, and zero-crossing rate to analyse and represent songs. This system aims to recommend music tailored to users' preferences by calculating the similarity between user preferences and song audio features. The accuracy of their system using artificial neural networks (ANN) is uncovered, with mean precision@10 of 0.148 and mean normalised discounted cumulative gain (nDCG) of 0.164, illustrating the effectiveness of their approach in recommending relevant songs to users. Positively, the system demonstrates efficiency in handling large databases, processing audio data, and accommodating various music genres. In essence, this content-based music recommendation system excels in its adaptability and database management capabilities (Niyazov et al., 2021).

A Siamese Neural Network (SNN) is used in Pulis and Bajada's (2021) music recommendation system to determine how similar two audio snippets are. Every clip is first pre-processed into a Mel-Spectrogram, which is then used as input for two Convolutional Neural Nets (CNNs) that are the same inside the SNN. In order to determine song similarity, the outputs of both CNNs are compared. This information is then used in a query-by-multiple-example (QBME) recommender system to find songs that match the user's tastes. The effectiveness of the system in recommending new music was demonstrated by an online survey, which found that 60.7% of 150 participants gave the SNN-based algorithm's recommendations a higher rating than the baseline system's, and that 55% of the recommended songs had received fewer than 1,500 listens. Some noteworthy benefits of the system are that it provides equal exposure to artists regardless of their level of popularity. On the other hand, some potential drawbacks include the need for a large dataset for SNN training and the possibility of recommending songs that are not in line with the user's preferences if the training data is not representative (Pulis & Bajada, 2021).

A content-based movie recommendation system that uses machine learning algorithms to provide consumers with tailored movie recommendations is presented by Katkam et al. (2023). Based on attributes like genre, cast, and director that are retrieved from movie metadata, the system applies the K-Nearest Neighbour (KNN) algorithm to find films that are comparable to the user's search. Furthermore, by differentiating between positive and negative reviews, sentiment analysis performed on IMDB reviews with the Naive Bayes classifier improves suggestions. Remarkably, the sentiment analysis scored 88% on test data and 92% on training data. Benefits include reducing the "cold start" problem that collaborative filtering techniques face and improving suggestion quality by utilising sentiment analysis. Nevertheless, the system's shortcomings result from its exclusive dependence on content-based filtering, which may cause it to ignore subtle user preferences that are detected by collaborative filtering methods. The system shows potential in personalised movie suggestions; the study proposes to improve recommendation accuracy and resilience by combining collaborative filtering algorithms and enlarging the dataset (Katkam et al., 2023).

In their study, Christina et al. (2023) conducted an investigation to optimise a content-based music recommendation system utilising the K-Nearest Neighbour (KNN) algorithm. Gathering song data and associated features from Spotify, they experimented with various distance metrics and feature selection threshold values to determine the most effective combinations. Evaluation was based on Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). Cityblock Distance emerged as the top-performing metric, yielding an average RMSE of 0.2257, while a feature selection threshold of 0.05 achieved optimal performance with an average MAE of 0.1279. The study underscores the significance of selecting appropriate distance metrics and feature selection thresholds to enhance the accuracy of music recommendation systems. However, limitations include the potential inadequacy of the Spotify dataset to represent the entire music landscape and the study's focus solely on distance metrics and threshold values, neglecting other factors like user contexts and temporal preferences that could further refine the recommendation system (Christina et al., 2023).

Sakti et al. (2022) present a Music Recommendation System centred on Content-Based Filtering Method employing the Euclidean Distance Algorithm. Utilising James Russel's Circumplex model to categorise user moods, the system calculates the Euclidean distance between songs to offer the top 10 recommendations closest to the user's mood input. Evaluation via NDCG (Normalised Discounted Cumulative Gain) yielded an average value of 0.95235, indicating highly relevant music recommendations. Pros include personalised recommendations based on user mood and the utilisation of a well-defined mood model. However, limitations include the system's reliance on content-based filtering, potentially missing collaborative filtering benefits, and the subjectivity of user input for mood categorization, suggesting continuous updates and improvements may be necessary for sustained recommendation accuracy (Sakti et al., 2022).

Mao et al. (2022) introduce the Music Classification and Recommendation Network (Music-CRN), a convolutional neural network designed for content-based music classification and recommendation. Music-CRN excels in both tasks, achieving state-of-the-art performance in music spectrogram classification with top-1 accuracy of 68.01% and top-3 accuracy of 89.37%, surpassing previous methods like MusicCNNs. Additionally, on full-scale music classification, Music-CRN attains the highest top-1 accuracy of 77.3% across various music genres. For music recommendation, Music-CRN computes cosine similarity between music tracks using feature vectors extracted from spectrograms, outperforming other content-based methods with top-1 recommendation accuracy of 71.50% and top-3 accuracy of 84.65%. While the model effectively addresses the cold-start issue of collaborative filtering methods and showcases superior feature extraction capabilities, limitations include the lack of publicly available dataset and insufficient detail regarding dataset composition. Overall, Music-CRN presents a promising solution for content-based music classification and recommendation tasks (Mao et al., 2022).

Summary Table

Research Paper	Author	Year	Method	Pros	Cons
Music similarity measurement and recommendation system using convolutional neural networks	Mohamadreza Sheikh Fathollahi and Farbod Razzazi	2021	Content-Based Recommendation System using convolutional neural networks (CNNs)	-Utilised deep learning techniques for accurate music genre classification. -Extracted features from CNN layers for precise similarity measurement.	-Performance varied based on input features and output layers -Short music pieces led to lower accuracy, suggesting limitations in certain datasets.
Music	Yezi Zhang	2022	Content-based algorithm	-Personalization	-Cold Start Problem

Recommendation System and Recommendation Model Based on Convolutional Neural Network			and collaborative filtering algorithm based on Convolutional Neural Networks (CNN) to provide personalised music recommendations.	-Accuracy -Adaptability -Efficiency	-Data Processing Burden -Scalability
Enhanced Music Recommendation Systems: A Comparative Study of Content-Based Filtering and K-Means Clustering Approaches	Sayak Mukhopadhyay, Akshay Kumar, Deepak Parashar, Mangal Singh	2024	Content-based filtering and k-means clustering , uses an extensive Spotify dataset encompassing diverse song attributes like genre, tempo, and key1.	-Content-based filtering emphasises the mirroring of musical properties between target and recommended songs - k-means clustering orchestrates a comprehensive symphony of song groupings based on shared audio attributes	-Content-based filtering may not capture the complexity of user preferences over time -K-Means clustering struggles to personalise suggestions to unique user preferences
Hotel Content-Based Recommendation System	Kristian Wahyudi	2020	Content-based recommendation based on the content of the item, such as genre, artist, and user ratings	- Can capture the full range of a user's preferences by considering the preferences of similar users. -Can handle situations where there is limited	- Requires a large amount of user data to be effective. - May not provide personalised recommendations for users with unique preferences.

				content data available.	
Content-based Music Recommendation System	Aldiyar Niyazov, Elena Mikhailova, Olga Egorova	2021	Content-based recommendation based on ANN and analysing the features of songs, such as tempo, rhythm, and genre, to create a profile for each song	- Personalised recommendations - Scalability - Efficient	- Requires a large dataset - May not capture changes in user preferences
Siamese Neural Networks for Content-based Cold-Start Music Recommendation	Michael Pulis, Josef Bajada	2021	Siamese Neural Network	- Provides fairer exposure to all artists irrespective of their fame or popularity	- Potential misalignment, risks recommending songs not aligned with user preferences if training data does not represent user tastes accurately. - Data dependency
Content-based Movie Recommendation System and Sentimental analysis using ML	Saketh Katkam,Abhishek Atikam,Pallerla Mahesh,Mrunal Chatre,Shree Sai Kumar,Sakthidharan G R	2023	- Naive Bayes classifier - K-Nearest Neighbour	- Avoids cold start - Incorporating sentiment analysis from reviews provides an additional signal to improve recommendation quality.	- system relies solely on content-based filtering, which may not capture all the nuances of user preferences -
Distance Metric Analysis in Recommendation System Using	Christina, Febrian Sanjaya, Valentina	2023	- K-Nearest Neighbour	- Choosing suitable distance metrics and implementing feature selection	- The study only considered distance metrics and certain threshold values, and did not explore other factors

Content-Based Filtering Method	Tiara Cahyaningtyas, Ivan Sebastian Edbert, Derwin Suhartono			can enhance the accuracy of music recommendation systems.	like user contexts, temporal preferences, or social factors that could further improve the recommendation system.
Music Recommendation System Using Content Based Filtering Method with Euclidean Distance Algorithm	Sri Mega Sakti, Arif Dwi Laksito, Bety Wulan Sari, Donni Prabowo	2022	- Euclidean Distance Algorithm	- Provides personalised music recommendations based on user mood. - Achieved a high NDCG value, indicating the relevance of the recommendations	- Relies heavily on user input for mood categorization, which may be subjective. - May require continuous updates and improvements to maintain recommendation accuracy over time.
Music-CRN: an Efficient Content-Based Music Classification and Recommendation Network	Yuxu Mao, Guoqiang Zhong, Haizhen Wang, Kaizhu Huang	2022	- Music Classification and Recommendation Network (Music-CRN), a convolutional neural network model for content-based music classification and recommendation.	- State-of-the-art performance on both music classification and recommendation tasks - avoids the cold-start - Effective feature extraction from music spectrograms using a novel CNN architecture	- Model Complexity and Interpretability - Subjectivity in Genre Division

Critiques

The landscape of music recommendation systems is rich with innovations, as evidenced by a range of studies exploring various methodologies. Mohamadreza Sheikh Fathollahi and Farbod Razzazi delve into convolutional neural networks (CNNs) to pioneer a content-based recommendation system for music. While their approach demonstrates commendable accuracy in genre classification, reliance on short music pieces and inaccessible datasets may limit broader applicability. Similarly, Yezi Zhang's investigation merges content-based and collaborative filtering algorithms, harnessing the power of CNNs for personalised music recommendations. Despite its effectiveness in offering tailored suggestions, challenges such as the cold start problem and computational burden cast shadows on scalability and efficiency. These studies shed light on the evolving landscape of music recommendation, emphasising the need for robust, scalable, and interpretable models to navigate the complexities of user preferences and system requirements.

In a quest for enhanced recommendation systems, Sayak Mukhopadhyay and colleagues delve into the comparative study of content-based filtering and k-means clustering approaches. While their findings underscore the strengths of content-based filtering and clustering techniques, challenges persist in adapting to evolving user preferences and tailoring recommendations to unique tastes. Meanwhile, Michael Pulis and Josef Bajada introduce a novel approach leveraging siamese neural networks, aiming to address the cold start problem while providing fair exposure to all artists. However, risks of misaligned recommendations and data dependency pose notable concerns. These studies collectively highlight the nuanced interplay between recommendation algorithms, user preferences, and system scalability, underscoring the ongoing quest for personalised, accurate, and adaptable music recommendation systems in an ever-evolving digital landscape.

2.2. Analysis of selected tool with any other relevant tools

Tools comparison	Remark	Jupyter Notebook	Visual Code Studio	Streamlit
Type of license and open source license	State all types of license	BSD License, Open Source	MIT License, Open Source	Apache License 2.0
Year founded	When is this tool being introduced?	2001	2015	2019
Founding company	Owner	Project Jupyter	Microsoft	Streamlit, Inc
License Pricing	Compare the prices if the license is used for development and business/commercialization	Free	Free	Free
Supported features	What features that it offers?	Interactive computing, code execution, visualisation, documentation, collaboration	Syntax highlighting, IntelliSense (code completion), debugging, Git integration, extensions ecosystem	Automatic reactivity and live updates, Simple Python API for creating interactive web apps
Common applications	In what areas this tool is usually used?	Data exploration, data analysis, machine learning, scientific computing	General-purpose programming, web development, cloud development	Data exploration and analysis, Prototyping machine learning models, Building interactive dashboards

Customer support	How the customer support is given, e.g. proprietary, online community, etc.	Online community, official documentation, support forums	Online community, official documentation, support forums	Documentation, support forums, GitHub
Limitations	The drawbacks of the software	Limited to Python (though kernels for other languages are available), less robust debugging compared to full-fledged IDEs	Heavier resource usage, may have performance issues with large codebases	Limited customization options for advanced users requiring highly tailored solutions, Performance constraints for handling large datasets

2.3. Justify why the selected tool is suitable

Jupyter Notebook creates a dynamic computing setting ideal for creating a music recommendation system, allowing for exploration of various algorithms and data processing methods in a systematic way. The ability to incorporate rich media allows for music data visualisation and recommendation analysis using charts, graphs, and audiovisual content, while the feature to merge code, text, and visualisations in one document aids in detailing and explaining the system's implementation and results. Jupyter Notebook has a vast and engaged community that offers a wide variety of libraries, extensions, and resources, which can improve development options and promote teamwork.

Visual Studio Code (VS Code) is a lightweight and extremely adaptable code editor ideal for creating the backend elements of a music recommendation system. Its user-friendly interface, with syntax highlighting, code completion, and built-in debugging tools, simplifies development and increases efficiency. The wide range of extensions available for VS Code, particularly those designed for tasks in data science and machine learning, increase its capabilities and improve productivity. Furthermore, smooth incorporation with Git version control systems promotes teamwork in developing and managing project code, documents, and experiments, rendering VS Code perfect for creating and enhancing a music recommendation system.

Because of its simple Python API which enables quick prototyping and development without requiring a lot of front-end experience Streamlit stands out as the ideal option for the GUI of my music recommendation system. A dynamic user experience is created by its automatic reactivity and live updates which guarantee real-time interaction with recommendations. I can quickly integrate machine learning algorithms straight into the interface and keep a consistent workflow thanks to the tools smooth integration with data science tools. Furthermore the process of integrating my recommendation system into live environments is made easier by Streamlits flexible deployment options and robust community support which also give users access to a multitude of resources for cooperation and troubleshooting. All things considered Streamlits ease of use interactivity and strong ecosystem make it a flexible and useful tool for creating an understandable and captivating user interface for my music recommendation system.

3. Methodology

3.1. Description of dataset

The dataset that is used for the music recommendation is obtained online through kaggle. The dataset was uploaded publicly by a user named “SHRIRANG MAHAJAN”, inside the dataset contains songs, artists names, links to songs and lyrics which is suitable for this recommendation system project. The file was uploaded as a csv file and there are 643 unique values for the artist column, 44824 unique values for the song column, 57550 unique values for the link column, and 57494 unique values for the text column. Overall, the dataset contains sufficient information to be used in this project.

In the preprocessing stage, a series of steps were implemented to refine the dataset for effective analysis and modelling. The procedure started by using the re module in Python to apply regular expressions for matching patterns. More precisely, non-alphabetic symbols were replaced with a space, eliminating any unnecessary characters that could cause interference in the dataset. Afterwards, all text was changed to lowercase uniformly in order to maintain consistency for future tasks like tokenization and removal of stop words. Tokenization involves dividing the text into separate words or tokens, to simplify the textual data into basic components. After that, a collection of English stop words from the NLTK corpus was used to remove typical, uninformative words that might affect analysis outcomes. This stage assists in directing the analysis towards the most significant information present in the text. In the end, the text that had been prepared beforehand was put back together into logical sentences or phrases, all set for additional examination or analysis. In general, this careful preprocessing method guarantees that the data set is improved and prepared for future analysis stages, like the development of a music recommendation system.

3.2. Applications of the algorithm(s)

In this recommendation system, the GloVe algorithm plays a central role in crafting vector representations for songs based on their lyrical content. GloVe, short for Global Vectors for Word Representation, stands as a pivotal technique in natural language processing, tasked with generating word embeddings by analysing the global co-occurrence statistics of words within a given corpus. Here, a pre-trained GloVe model is employed, sourced from a file housing word embeddings previously trained on an extensive text corpus. Each word within a song's lyrics is then mapped to its corresponding vector representation using this pre-trained GloVe model. These resultant word vectors encapsulate the semantic essence of words, capturing their meanings in a continuous vector space.

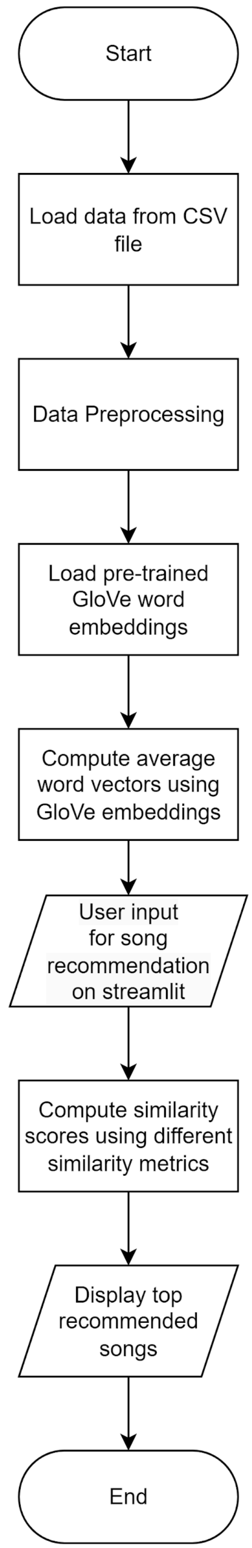
With the GloVe embeddings of song lyrics at hand, the recommendation system proceeds to gauge similarity between songs through diverse similarity metrics. Among these metrics, cosine similarity emerges as a prominent measure, evaluating the cosine of the angle between two vectors within a multi-dimensional space. Employed here, cosine similarity computes the resemblance between the vectorized renditions of songs, derived from their GloVe embeddings. Elevated cosine similarity scores denote a heightened affinity between songs, indicative of shared semantic traits within their lyrical content. By leveraging cosine similarity, the system adeptly identifies songs exhibiting significant thematic overlap with a given user input.

Beyond cosine similarity, the recommendation system embraces Jaccard similarity as another pivotal metric for assessing song similarity grounded in lyrical content. Jaccard similarity, a concept rooted in set theory, quantifies the likeness between two sets by measuring the ratio of their intersection to their union. Within this framework, Jaccard similarity quantifies the overlap in words present across song lyrics, with higher scores signalling heightened correspondence. Through Jaccard similarity, the system effectively captures semantic congruence among songs based on shared vocabulary, enriching its ability to discern thematic affinities.

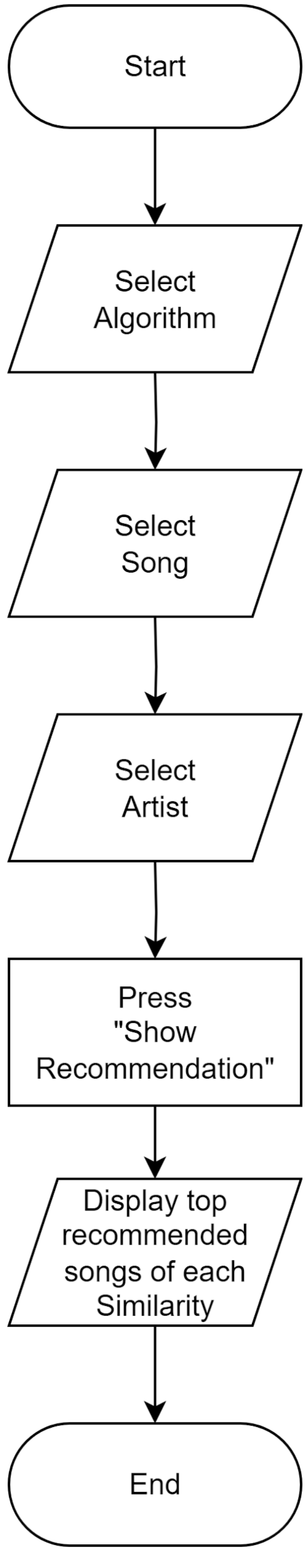
Furthermore, the recommendation system leverages the Pearson correlation coefficient to discern similarity between songs based on their GloVe-derived vector representations. The Pearson correlation coefficient, a statistical measure of linear correlation, gauges the strength and direction of the linear relationship between two variables. In this context, the coefficient is computed between the vector representations of songs, offering insights into their relational patterns within the vector space. By incorporating the Pearson correlation coefficient, the system unveils songs that exhibit congruent trends or patterns in their GloVe embeddings, complementing the insights gleaned from cosine and Jaccard similarity metrics.

3.3. System flowchart/activity diagram

3.3.1 Data Flowchart



3.3.2 User Interface Flowchart



3.4. Proposed test plan/hypothesis

Test Plan:

No	Test Case	Test Data	Case	Sample Input	Expected Results	Actual Results	Pass/Fail	Comments
1	Algorithm Selection	1.Word2Vec2.GloVe 3.Doc2Vec	1	Selects Word2Vec option	Algorithm gets selected successfully	Algorithm matched with the available options and gets selected successfully	Pass	
			2	Selects GloVe option	Algorithm gets selected successfully	Algorithm matched with the available options and gets selected successfully	Pass	
			3	Selects Doc2Vec option	Algorithm gets selected successfully	Algorithm matched with the available options and gets selected successfully	Pass	
			4	Type in anything else that is not valid	Nothing shows up to let user select	Nothing shows up that matches the input for selection	Pass	If “test” was entered, no algorithm would show up to let the user select because it’s not valid.

2	Song selection after a valid algorithm has been selected	1.”Right now” 2.”test123” 3. “love”	1	Select Glove then type “right now” for the song section	“Right now” should be available to be selected as a song	“Right now” is found and can be selected	Pass	“Right now” is one of the available songs in the dataset
			2	Select GloVe for the algorithm then type “test123” in the song section	No song would match the input entered	“test123” is not found	Pass	“test123” is not a song in the dataset
			3	Select GloVe for the algorithm then type “love” in the song section	The result will show songs that contains “love” in the song’s name	Result shows songs that contain “love” in the song’s name	Pass	“Lovers” and “Love me” will show up in the list
3	Artist Selection	1. One Direction 2. Two Direction 3. blank	1	Select Glove algorithm, select “right now” for the song section, then enter “One Direction”	“One Direction” will appear for selection	“One Direction” appeared in the list for selection	Pass	There is a song called “Right Now” by “One Direction”
			2	Select Glove algorithm, select “right now” for the song section, then enter	“Two Direction” will not appear for selection because it does not exist	“Two Direction” did not appear in the list for selection	Pass	There does not exist a song called “Right Now” by “Two Direction”

				"Two Direction"				
			3	Select Glove algorithm, select "right now" for the song section, then leave the artist section blank	Artists that have a song named "Right Now" will appear	Artists that have a song named "Right Now" appeared	Pass	Artist named "One Direction" and "Korn" will appear on the selection list
4	Show Recommendation	1.Glove, "Right Now", "One Direction" as the algorithm, song, and artist	1	Select Glove, "Right Now", "One Direction" as the algorithm, song, and artist, then click recommend	Recommend songs based on cosine similarity, jaccard similarity, and pearson coefficient similarity	Songs recommended based on cosine similarity, jaccard similarity, and pearson coefficient similarity	Pass	Top 10 of each similarity. (Cosine Similarity, Jaccard Similarity, Pearson Correlation Coefficient)
5	Song Link	1.Glove, "Right Now", "One Direction" as the algorithm, song, and artist	1	Click recommend then click the link of one of the songs recommended	Take user to the song page on spotify website	User gets taken to the song page on spotify website	Pass	Song can be played on the song page
6	Time taken	1.Glove,	1	Click	Time taken for	Time taken	Pass	Time taken is

	for algorithm recommendation	"Right Now", "One Direction" as the algorithm, song, and artist		Recommend	the selected algorithm recommendation will be displayed	for the selected algorithm recommendation is displayed		shown in seconds
7	Total running time	1.Glove, "Right Now", "One Direction" as the algorithm, song, and artist	1	Click Recommend	Total running time will be displayed at the bottom of the page	Total running time is displayed at the bottom of the page	Pass	Total running time is shown in seconds

4. Result

4.1. Results

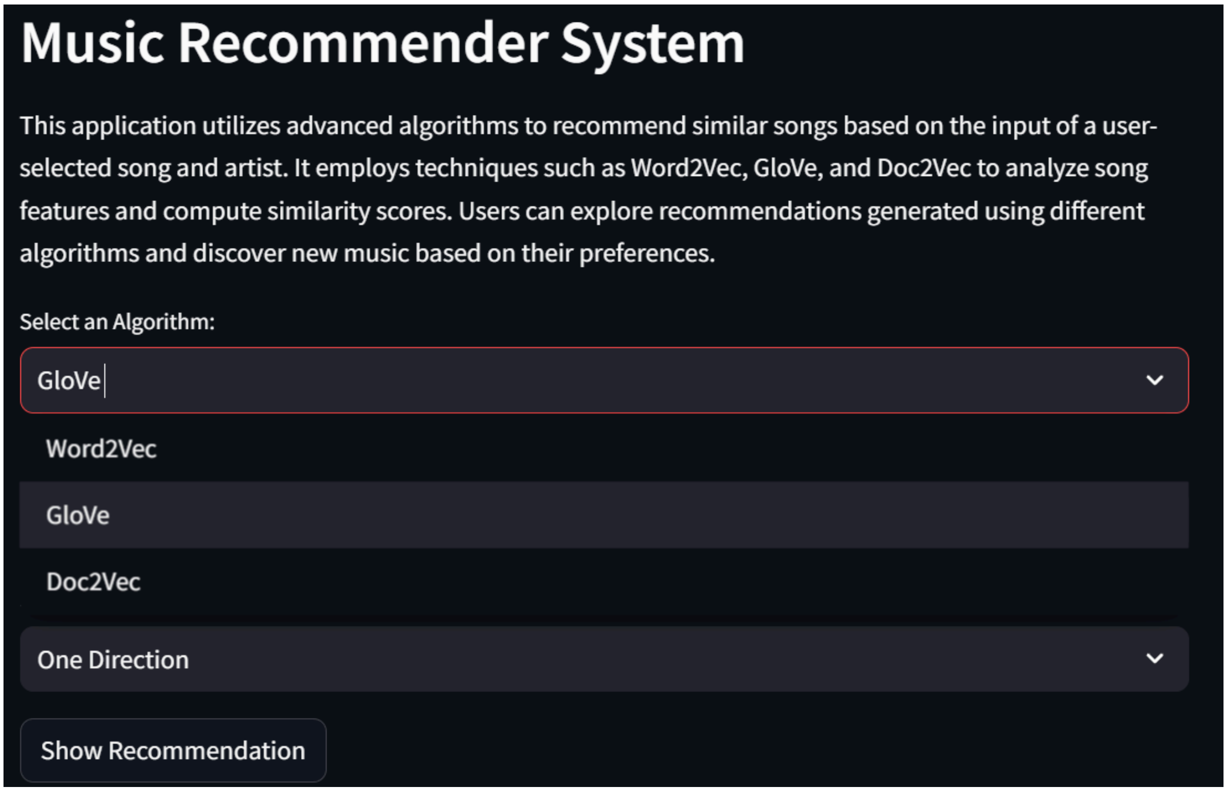


Figure 4.1: Algorithm Selection, selecting GloVe as the algorithm

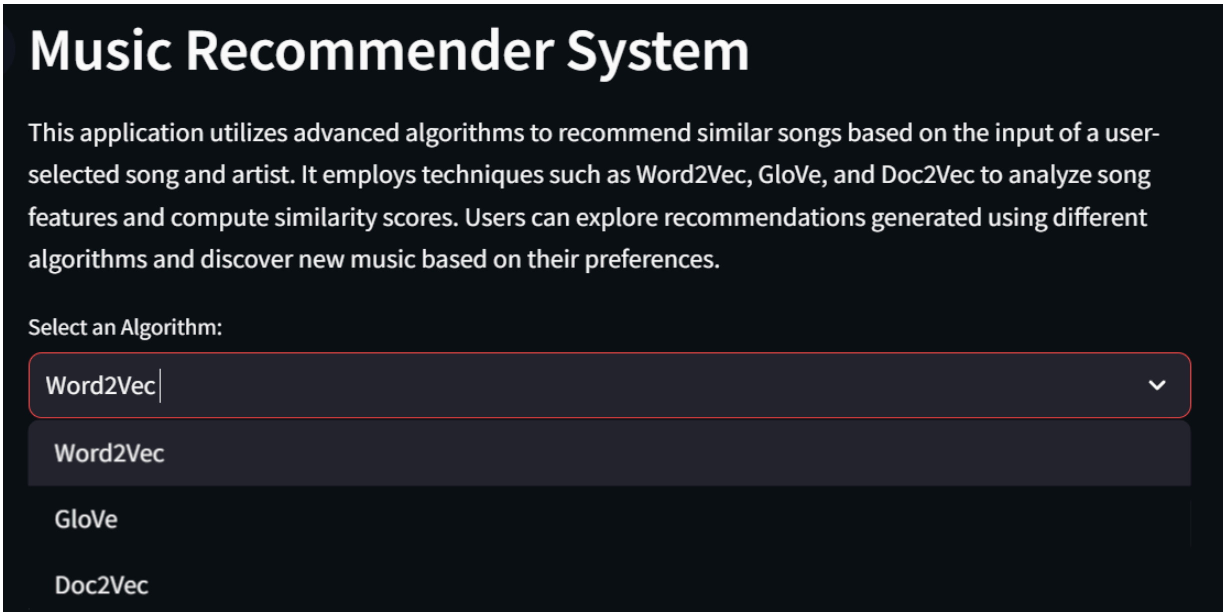


Figure 4.2: Algorithm Selection, selecting Word2Vec as the algorithm

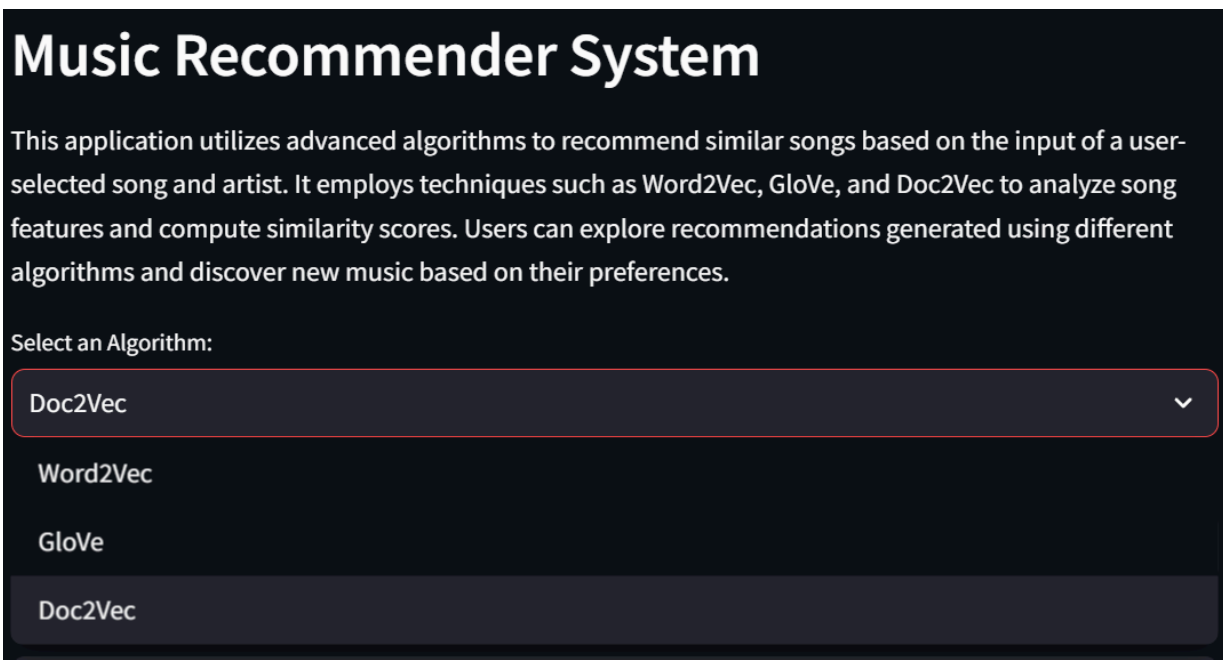


Figure 4.3: Algorithm Selection, selecting Doc2Vec as the algorithm

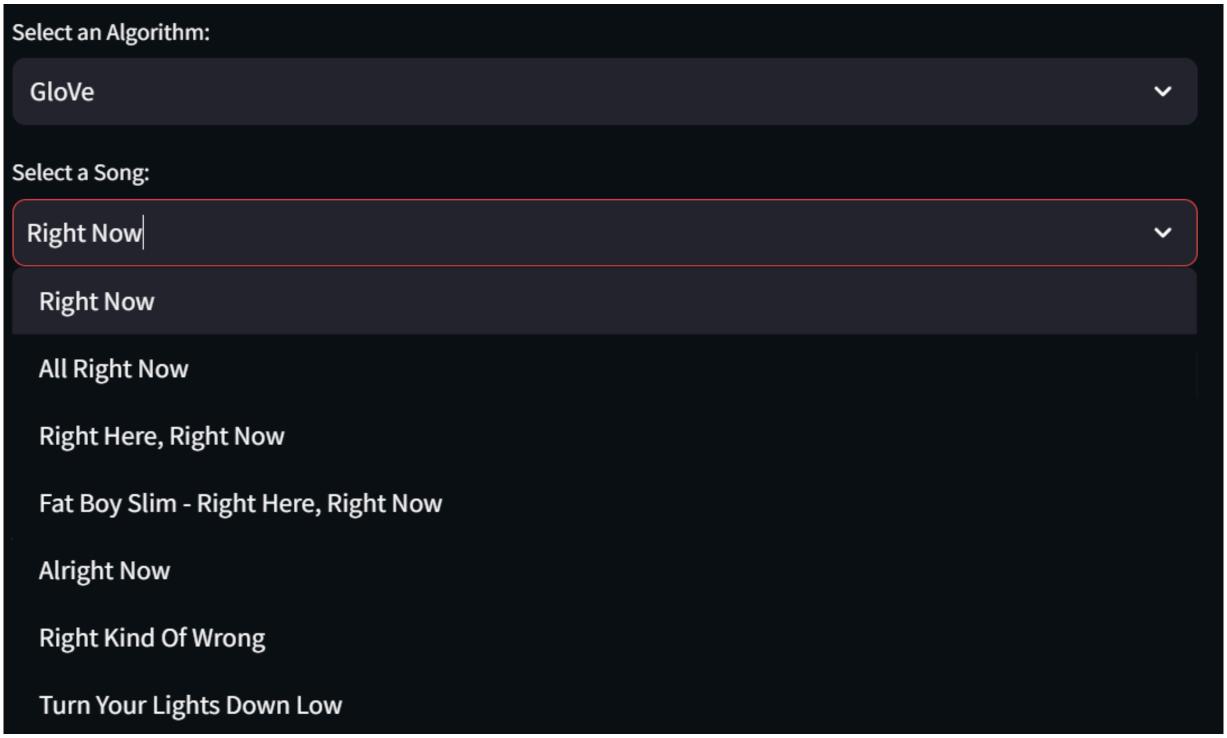


Figure 4.4: Song Selection, “Right Now” is available to be selected because it is one of the available songs in the dataset

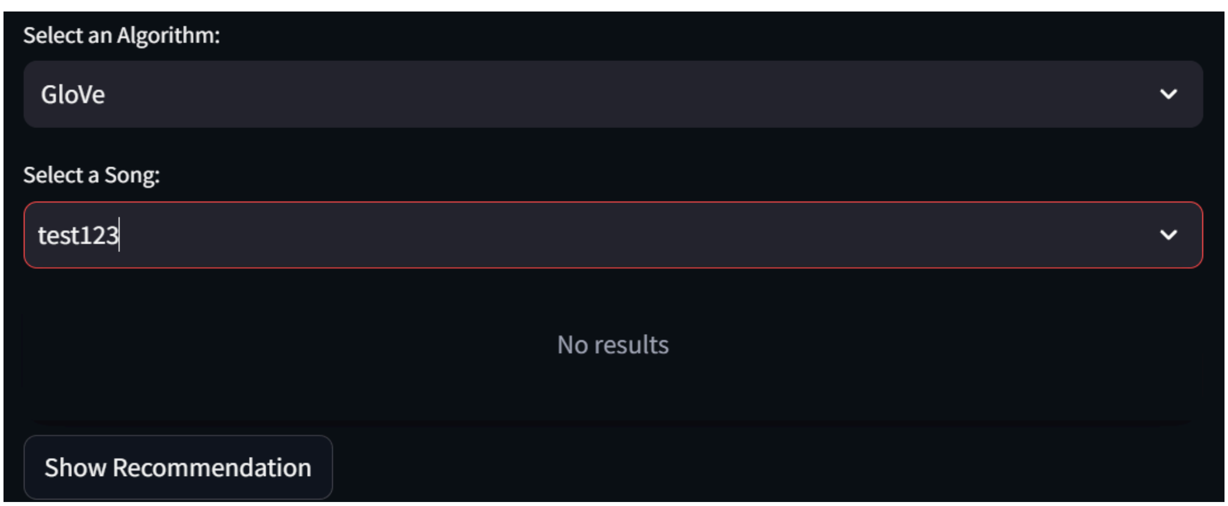


Figure 4.5: Song Selection, “test123” is not available to be selected because it is not one of the available songs in the dataset

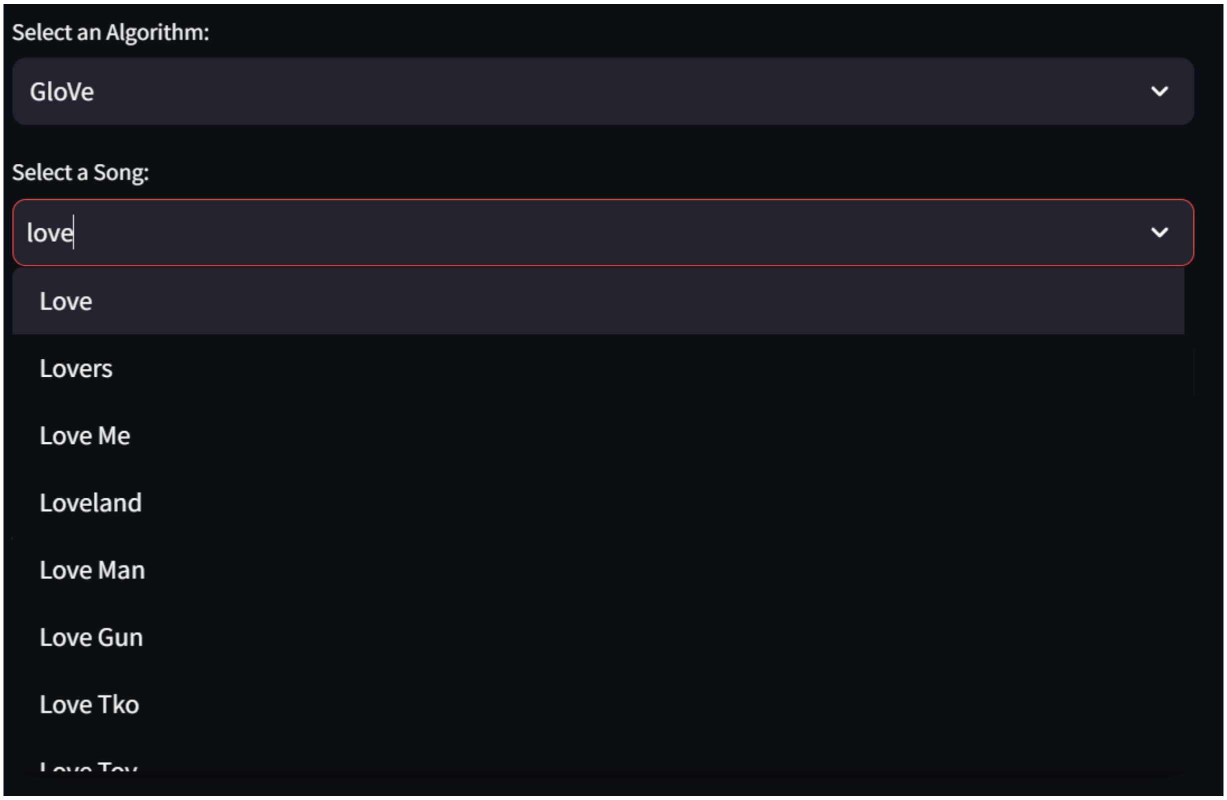


Figure 4.6: Song Selection, “love” is entered and the result shows songs that contain “love” in the name.

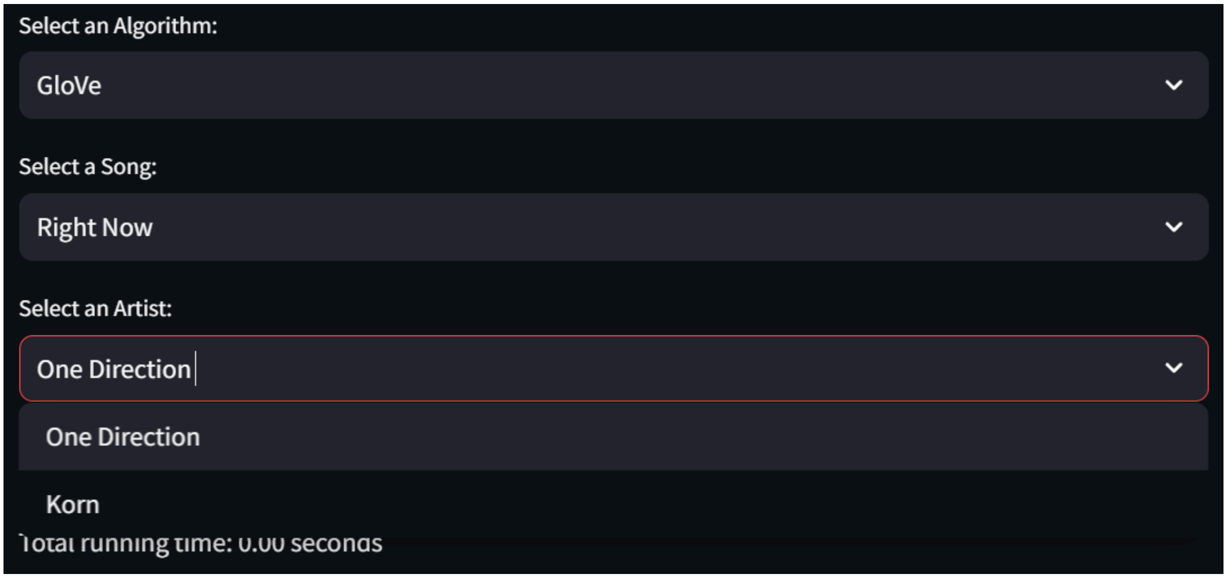


Figure 4.7: Artist Selection, “One Direction” is entered and the result shows “One Direction” as a select option

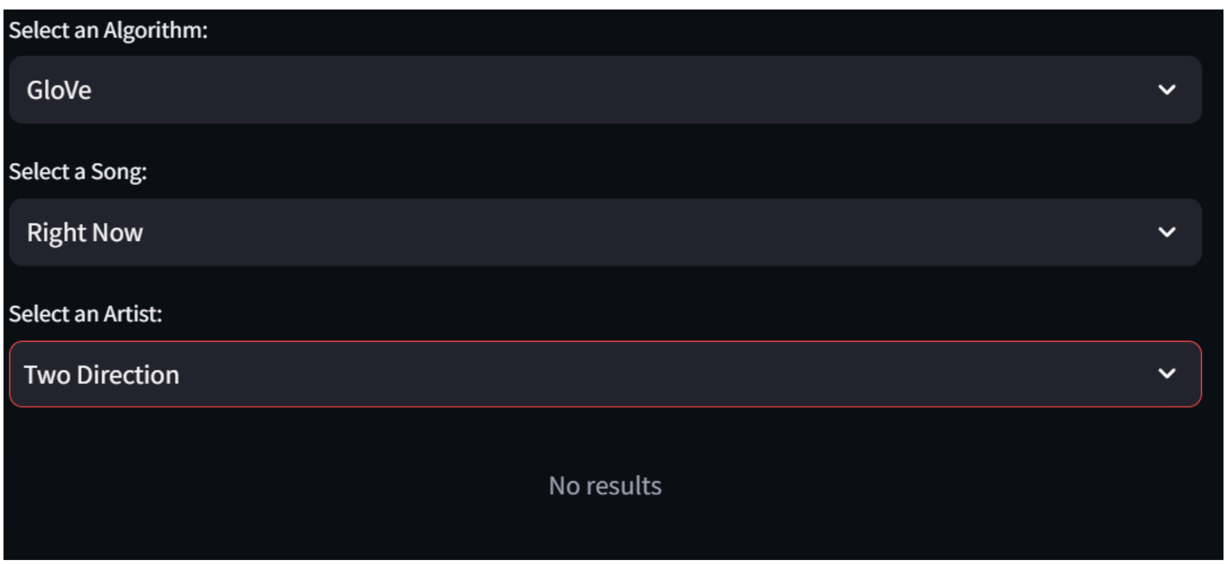


Figure 4.8: Artist Selection, “Two Direction” is entered and the result shows no results because artists that have a song called “Right Now” can’t be found

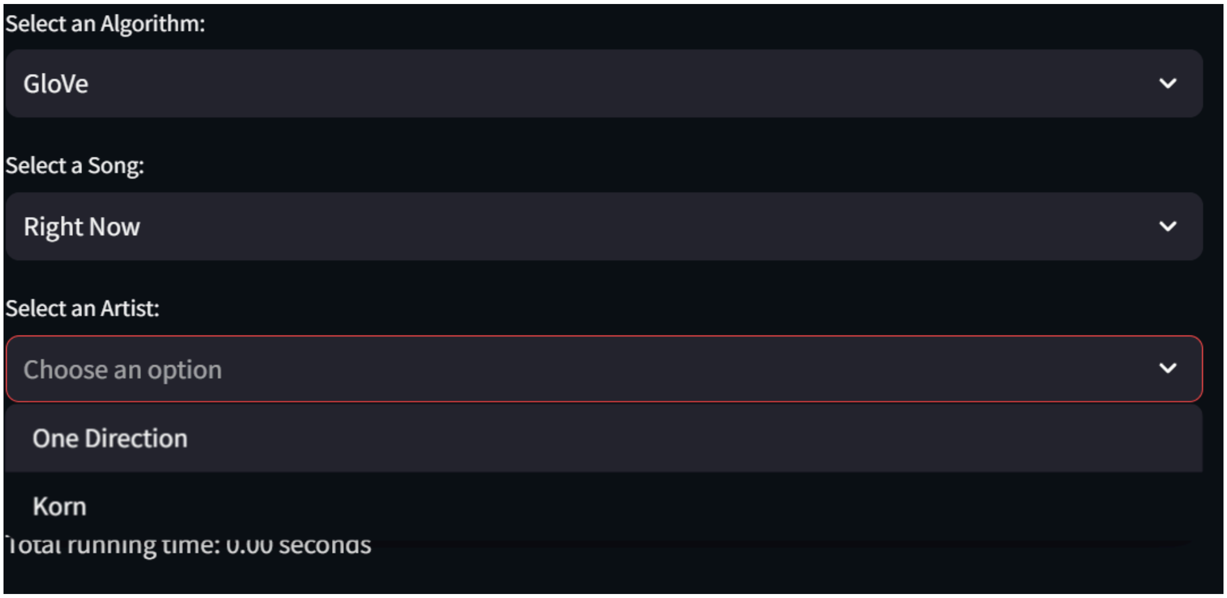


Figure 4.9: Artist Selection, artist is left blank and all available artists will be shown

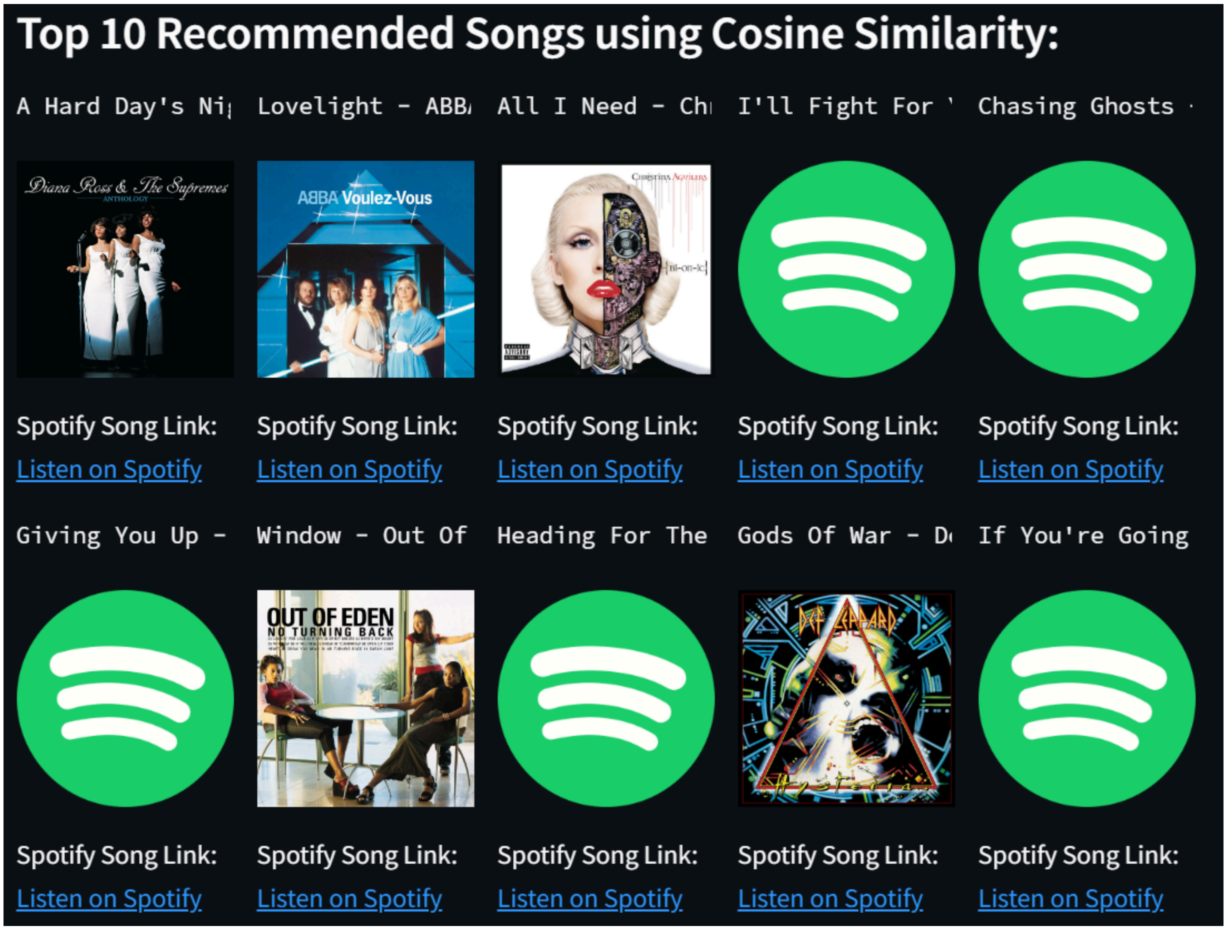


Figure 4.10: Show Recommendation, songs recommended based on cosine similarity is shown

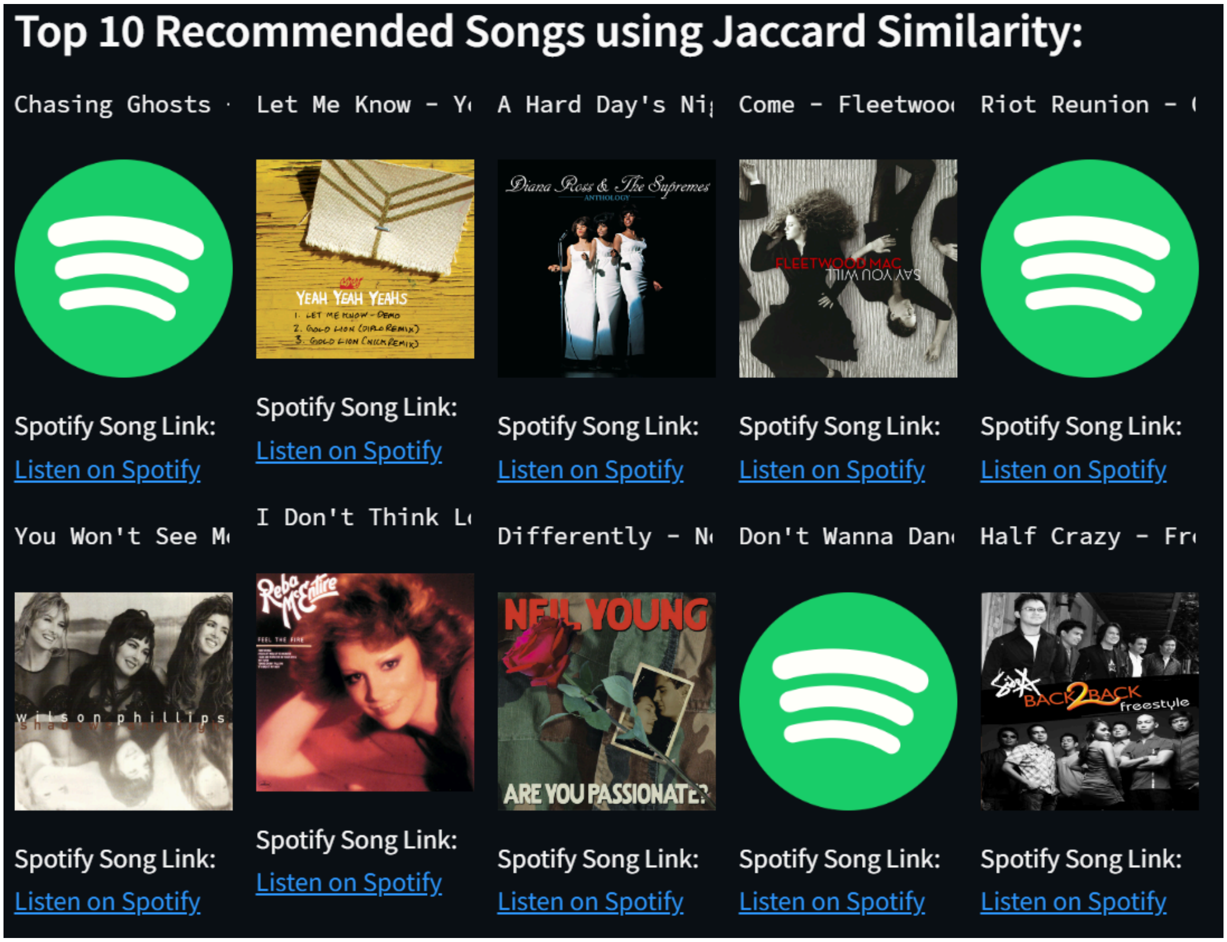


Figure 4.11: Show Recommendation, songs recommended based on Jaccard similarity is shown

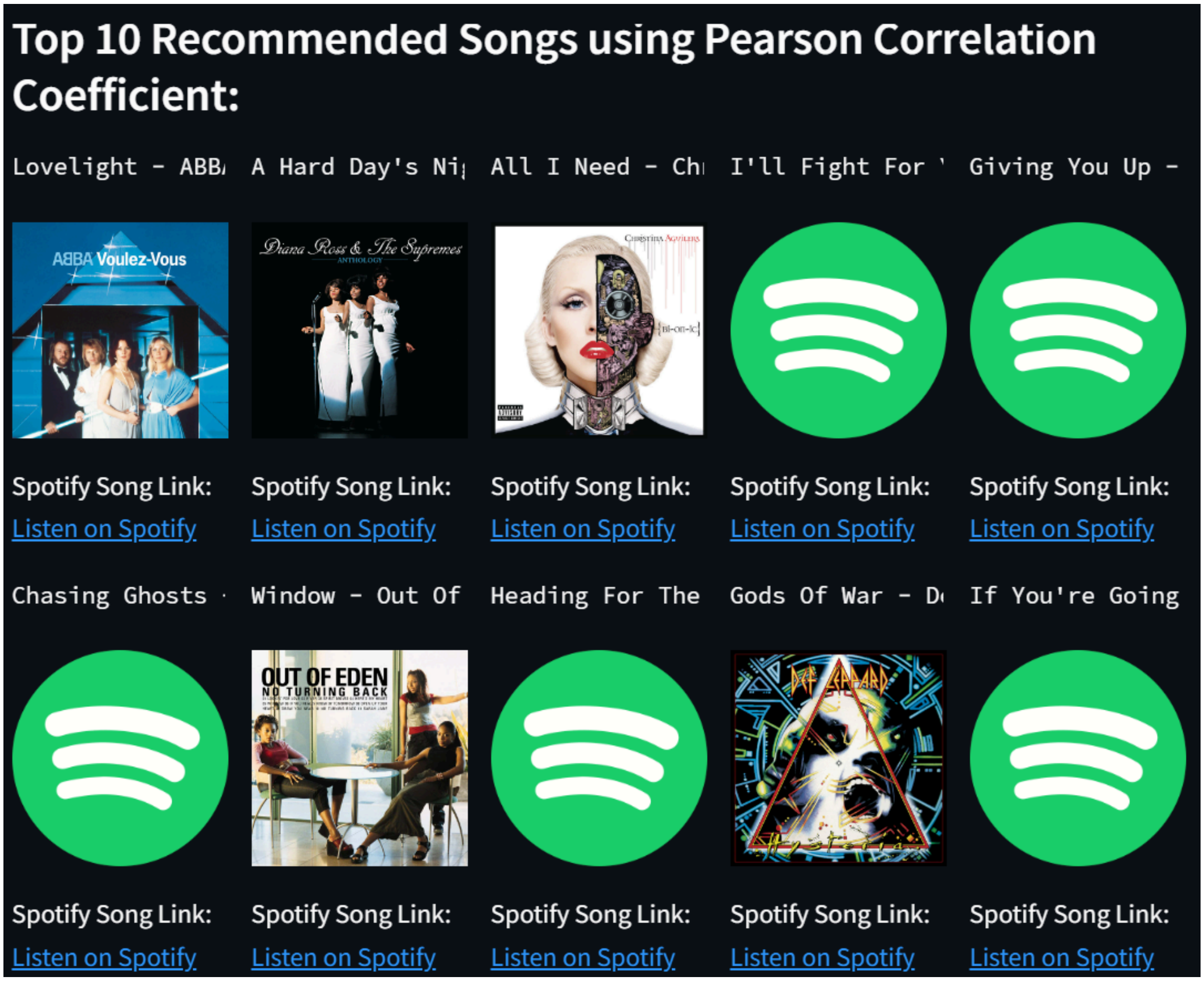


Figure 4.12: Show Recommendation, songs recommended based on Pearson Coefficient Correlation is shown

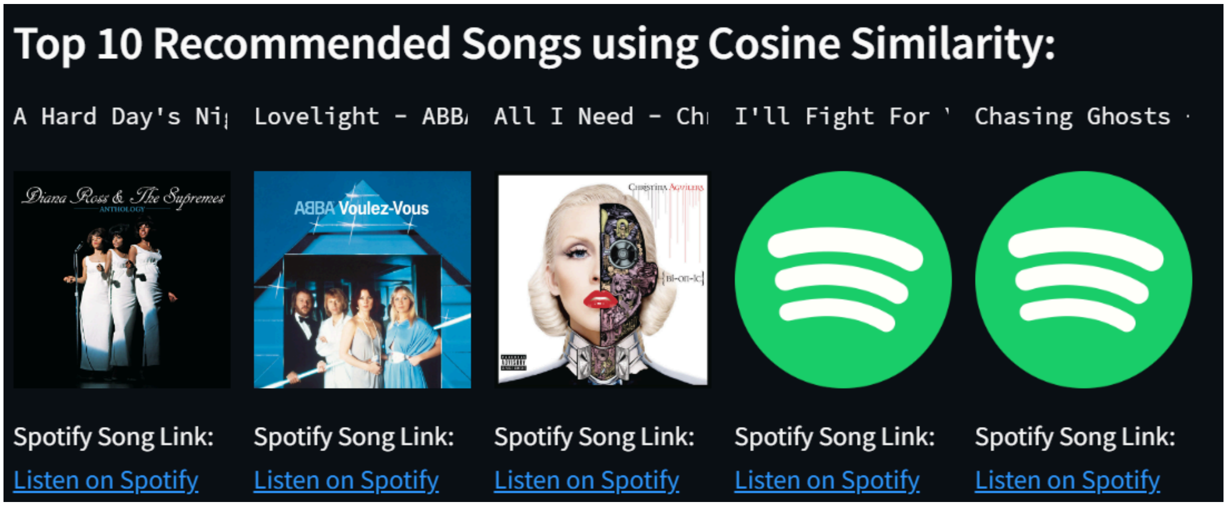


Figure 4.13: Song Link, The link of the first song recommended on the far left is clicked to go to the website.

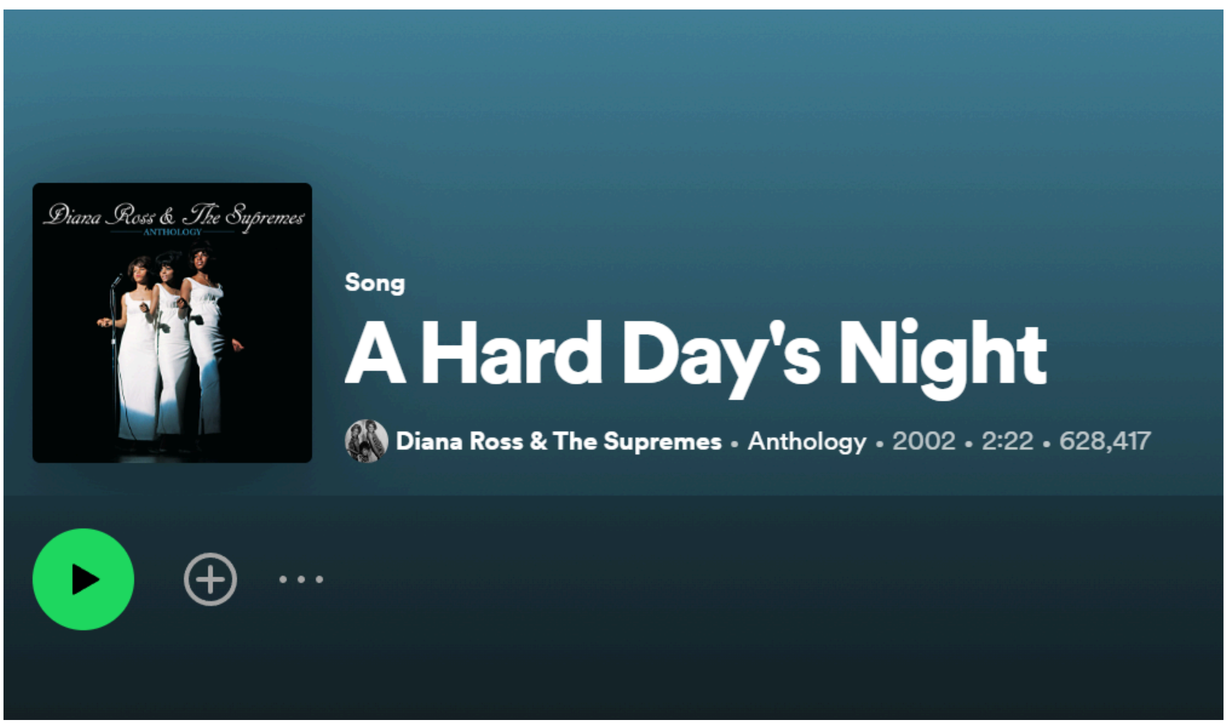


Figure 4.14: Song Link, user gets taken to the song website following after figure 4.13

Figure 4.15: Time taken for algorithm recommendation is shown after recommendation is clicked

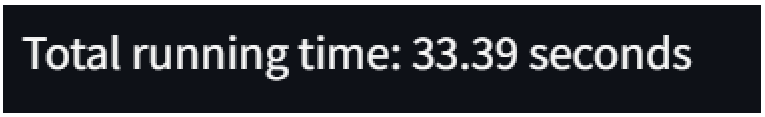


Figure 4.16: Total running time is shown after recommendation is clicked

4.2. Discussion/Interpretation

After going through all the algorithms to recommend songs based on one specific song. The GloVe algorithm appears to be the one algorithm that has the highest of all three similarity scores compared to other algorithms. The top one recommended song that got recommended using GloVe algorithm in figure 4.13 has a cosine similarity score of 0.9986, jaccard similarity score of 0.1803, and pearson correlation coefficient score of 0.9986. The one algorithm that did the worst performance based on similarity score is Doc2Vec and Word2Vec takes the second spot for the best performance algorithm. It also seems like the jaccard similarity metric did not do so well for the recommendation system. The reason that it struggled might be due to Jaccard similarity not being well suited in sparse data handling. The time taken for GloVe algorithm recommendation in figure 4.15 shows a very short time, indicating that GloVe algorithm is pretty efficient and quick in recommending songs. The total running time in figure 4.16 comes down to below 1 minute which is just acceptable. Overall The music recommendation system performed pretty well under the utilisation of the GloVe algorithm.

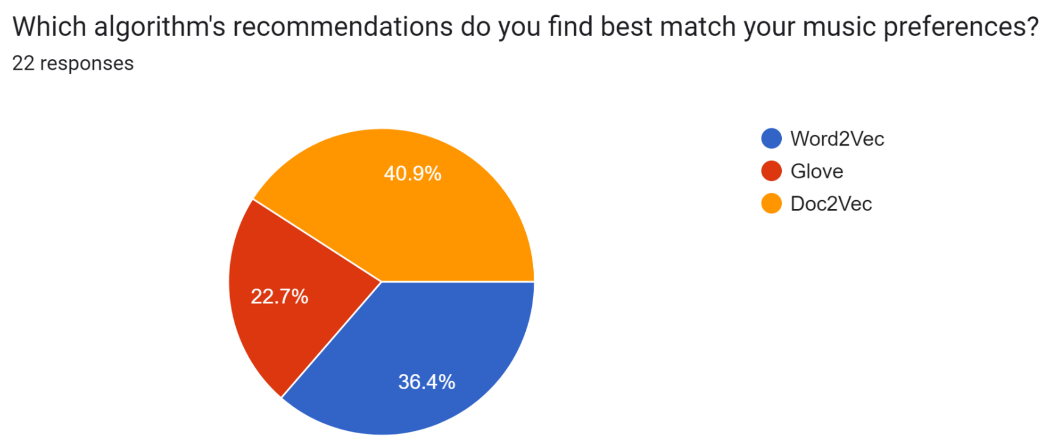


Figure 4.2.1: pie chart shows the percentage of algorithms chosen by the respondents.

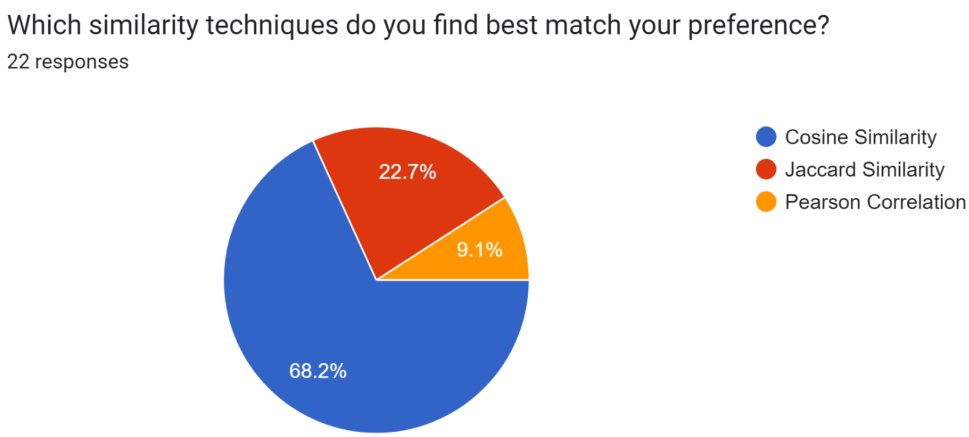


Figure 4.2.2: The percentage of similarity techniques that respondents find best match their preference.

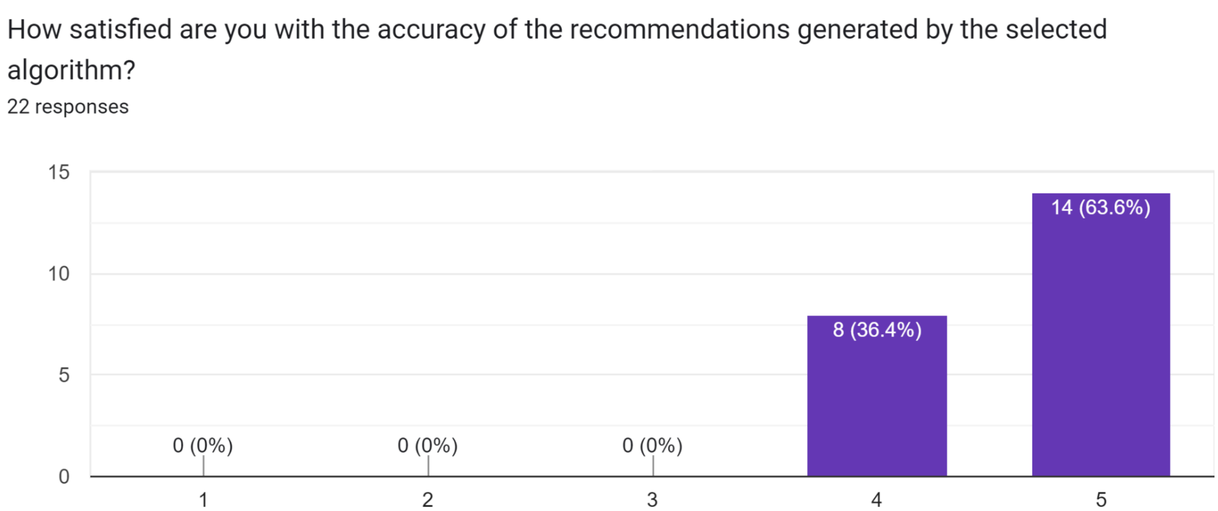


Figure 4.2.3: The percentage of how satisfied the respondents are with the accuracy of the recommendations generated by the selected algorithm. Majority thinks it's great.

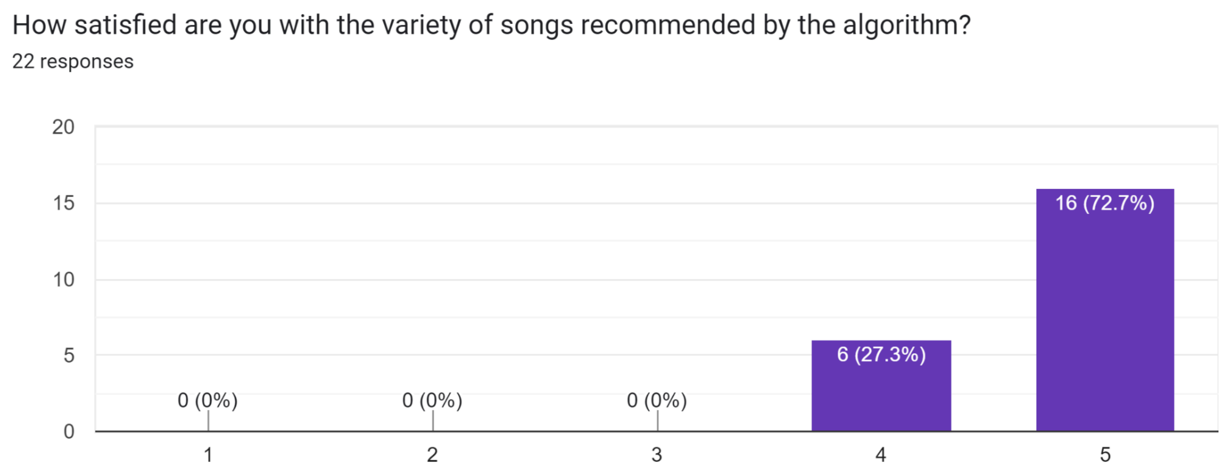


Figure 4.2.4: Majority of the respondents, 72.7% are satisfied with the variety of songs recommended by the algorithm

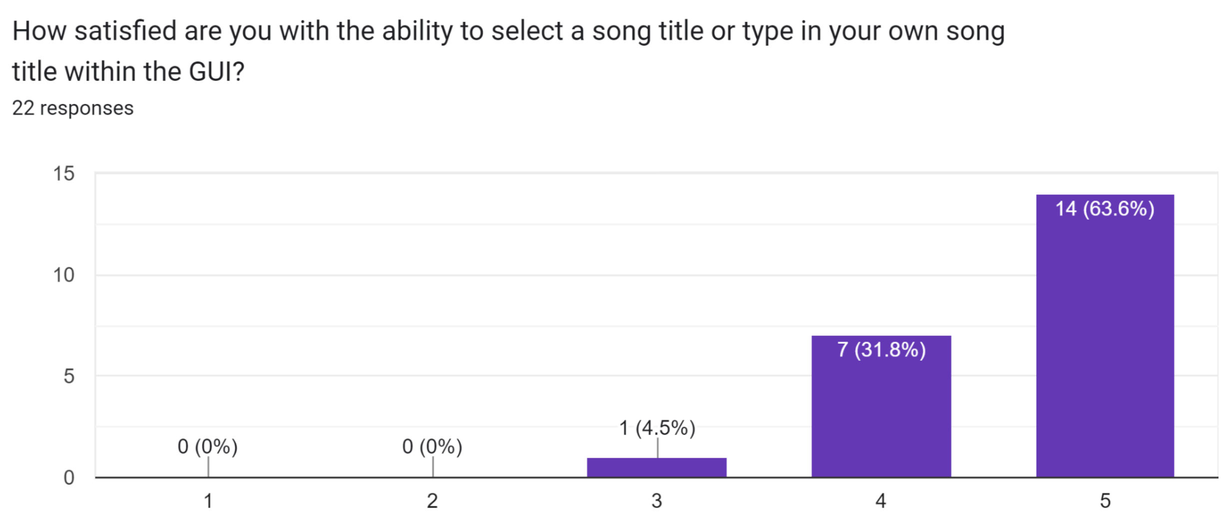


Figure 4.2.5: percentage of how satisfied the respondents are with the ability to select a song title or type in their own song title within the GUI. Majority of the respondents are satisfied.

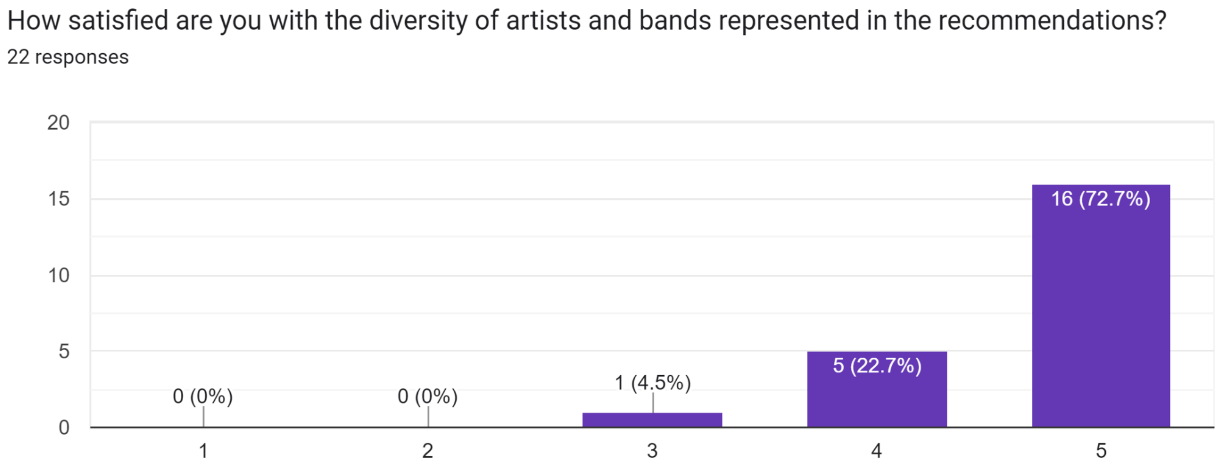


Figure 4.2.6: The result shows that most of the respondents are satisfied with the diversity of artists and bands represented in the recommendations

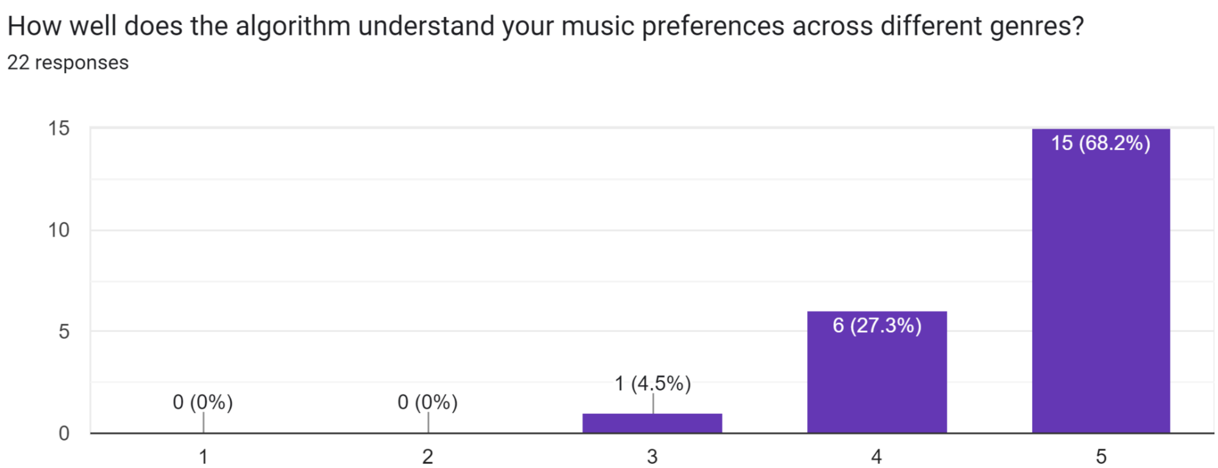


Figure 4.2.7: The result shows that the majority feels that the algorithm understand their music preferences across different genres very well

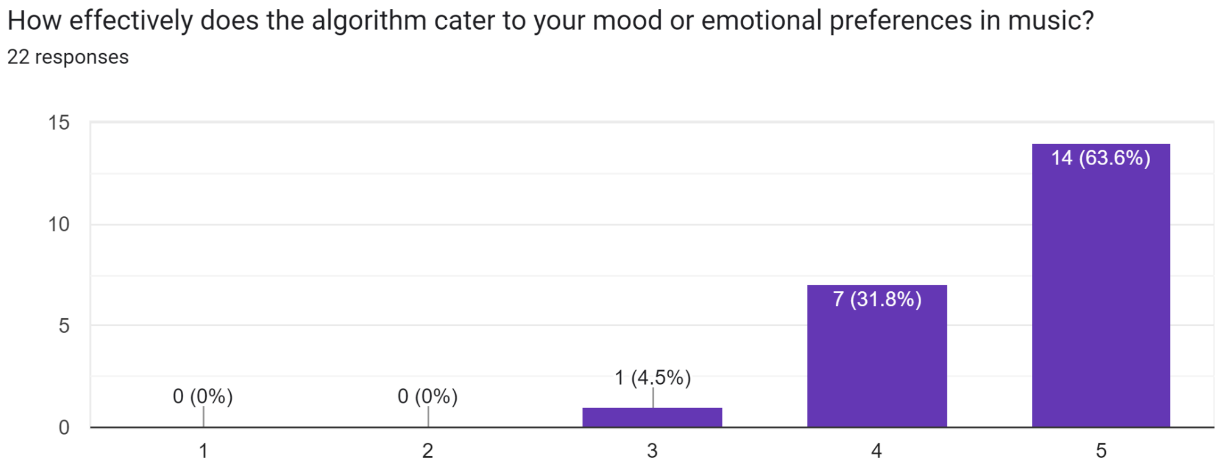


Figure 4.2.8: 63.6% of the respondents feel that the algorithm is very effective at catering to their mood or emotional preferences in music.

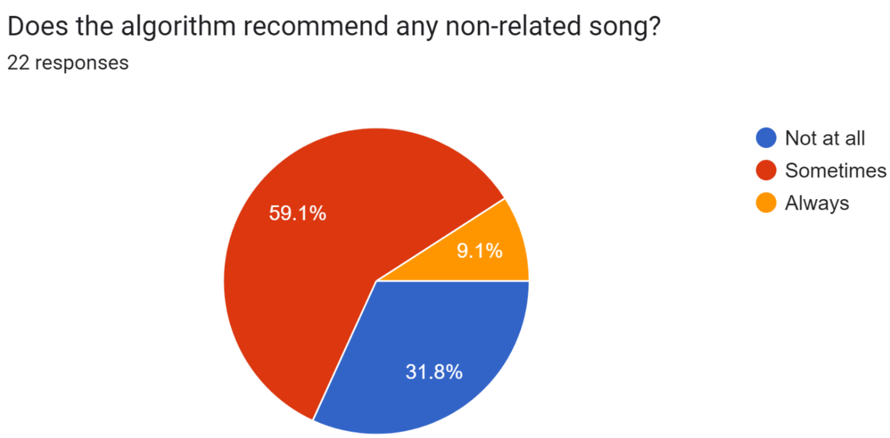


Figure 4.2.9: Over half of the respondents feel that the algorithm recommends non-related songs sometimes, 9.1% of the respondents feel that it is always, while 31.8% of the respondents never feel so.

Based on the google form questionnaire that we have sent out to our prototype testers, some useful data was obtained. As of now, the majority of the respondents are in the opinion that Doc2Vec algorithm's recommended songs match their preference the most based on figure 4.2.1, it is a whopping 40.9% for Doc2Vec, 36.4% for Word2Vec, and 22.7% for GloVe. In hindsight, it seems like having a higher similarity score does not necessarily mean the recommended song would match the user's preference. According to respondents, most of the respondents feel that cosine similarity is the best technique used to recommend songs that would suit their preference. With most of the respondents rated 4 or 5 out of 5, the respondents seem to be very satisfied with the accuracy of the recommendations generated by their selected algorithm. Also from the statistics, a large part of the respondents feels satisfied about the variety of songs recommended by the algorithm, the diversity of artists and bands represented in the recommendations, how well the algorithm understands their music preferences across different genres and how effectively the algorithm caters to their mood or emotional preferences in music even though most of them find that the system recommend them non-related songs sometimes.

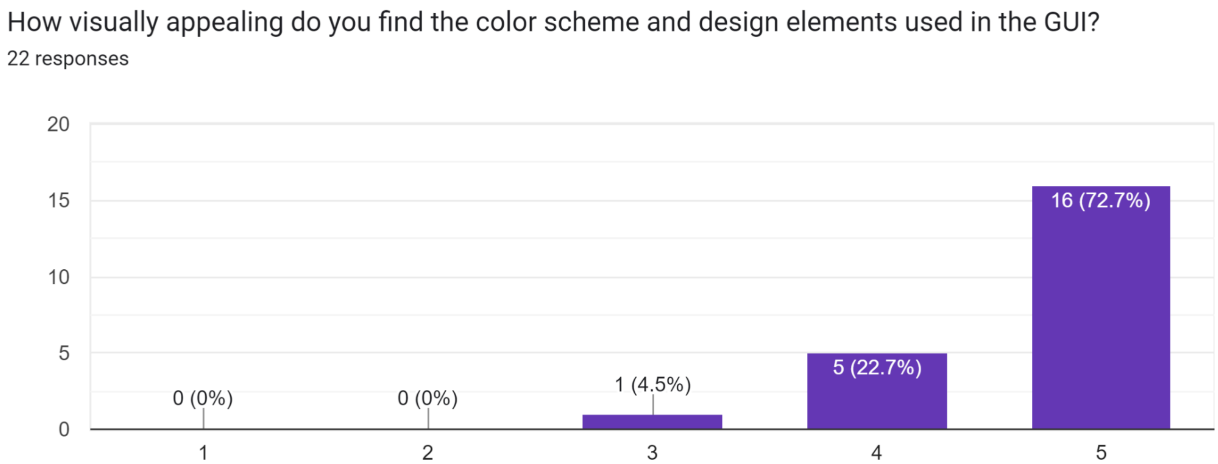


Figure 4.2.10: The GUI looks to be appealing to the majority of the respondents.

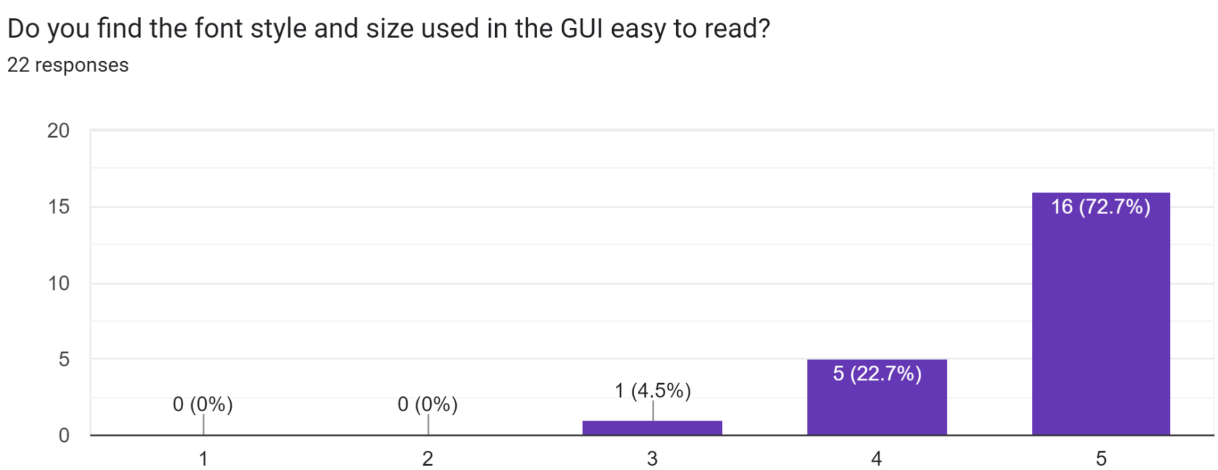


Figure 4.2.11: Percentage of how respondents think about the GUI font style and size

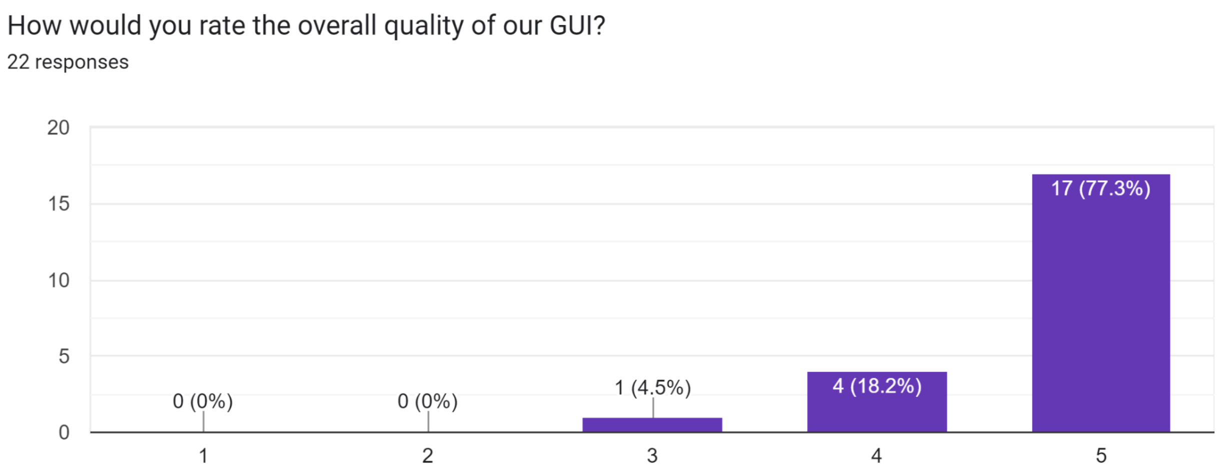


Figure 4.2.12: Majority of the respondents feel that the overall quality of the GUI is perfect.

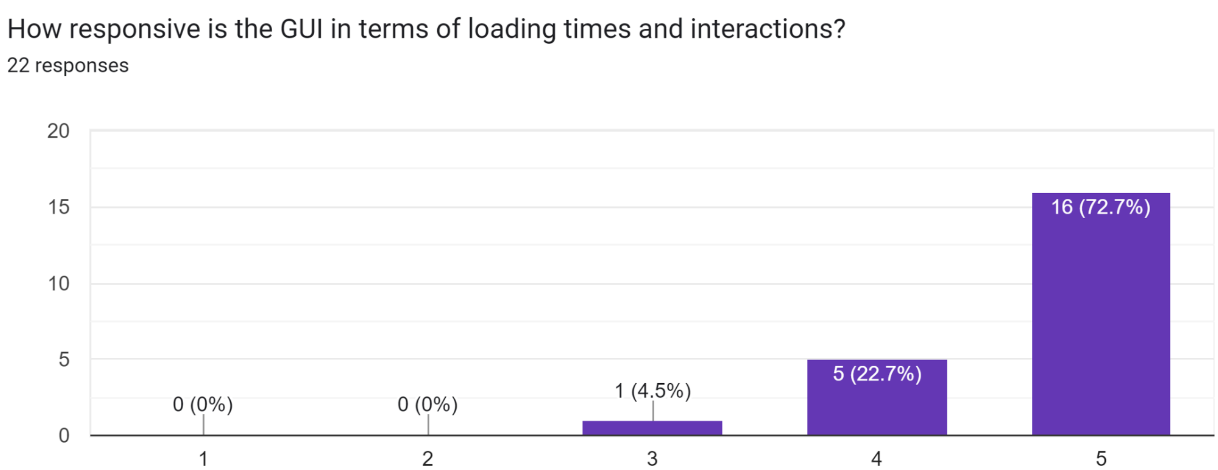


Figure 4.2.13: Most of the respondents are in the opinion that the responsiveness of the GUI in terms of loading times and interactions is just perfect.

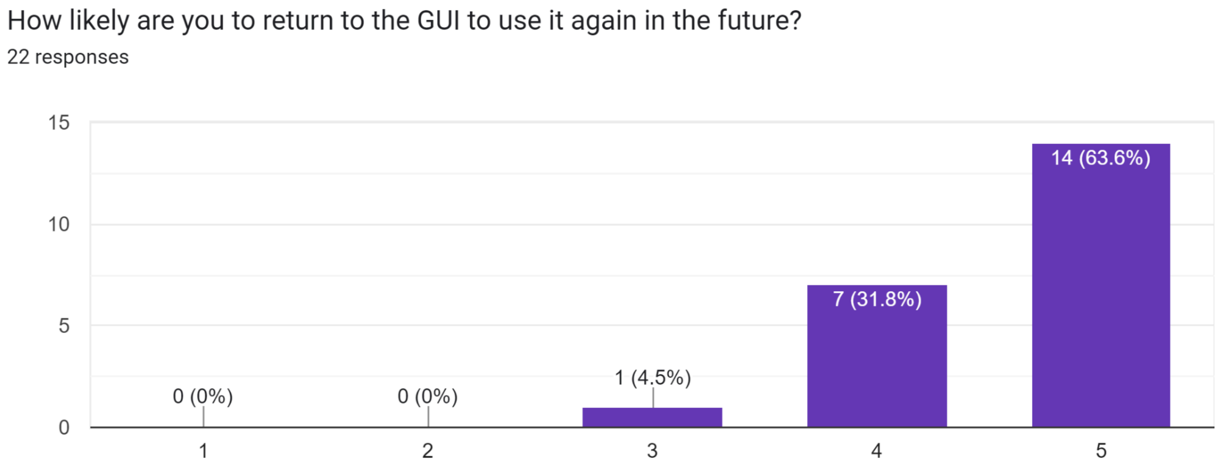


Figure 4.2.14: More than 60% of the respondents would return to the GUI to use it again in the future.

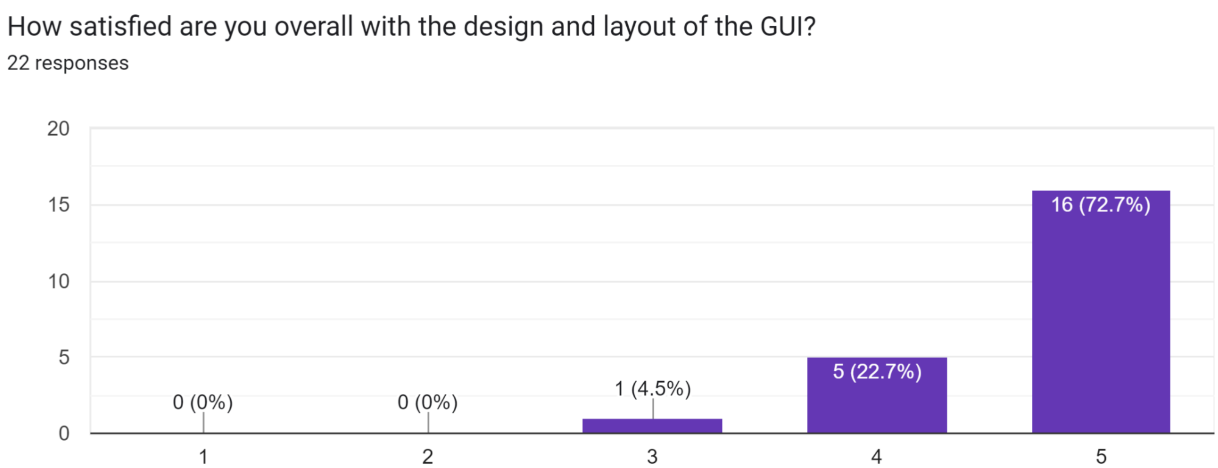


Figure 4.2.15: Most of the respondents feel very satisfied overall about the design and layout of the GUI.

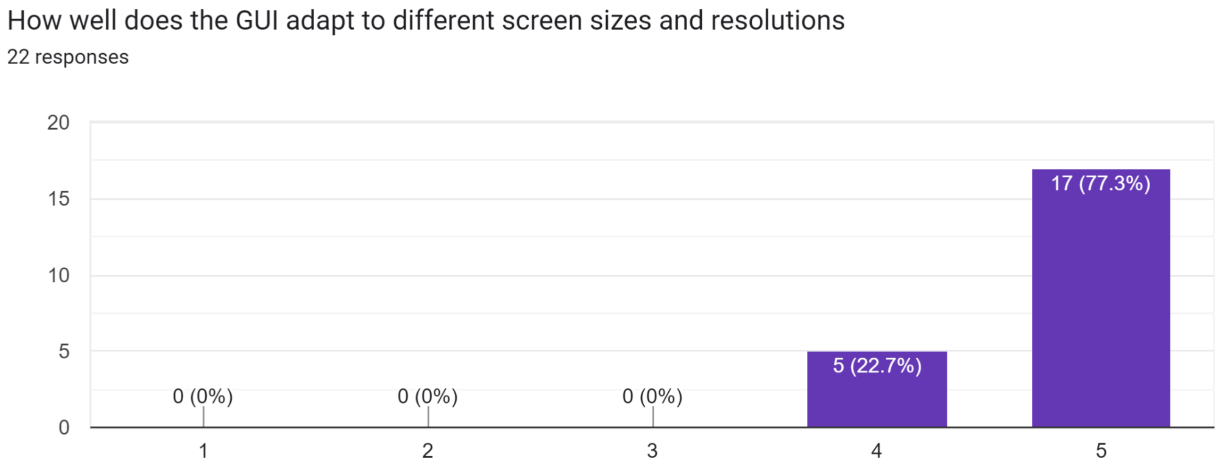


Figure 4.2.16: It is clear that the GUI adapts to different screen sizes and resolutions of the respondents.

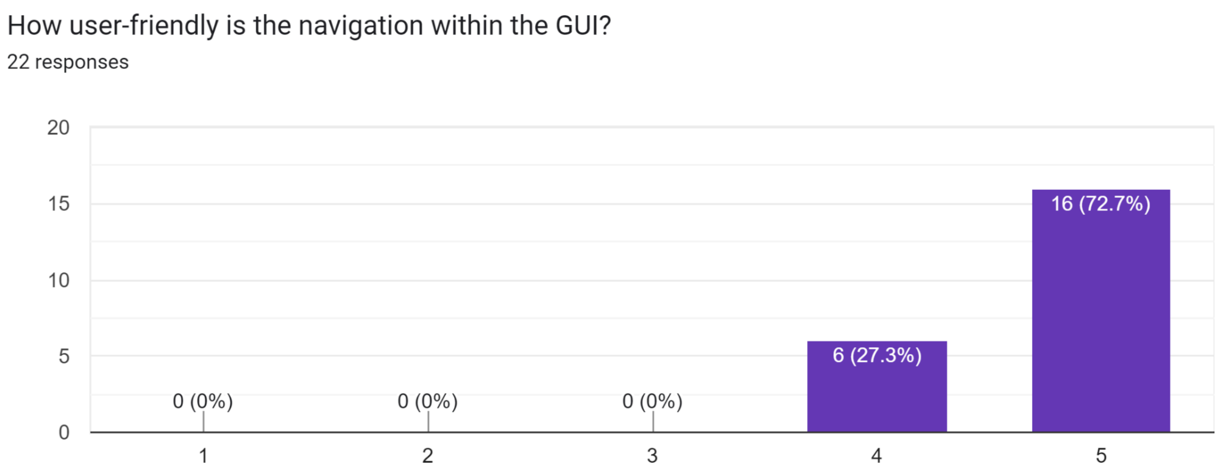


Figure 4.2.17: Majority feels the navigation within the GUI is user-friendly

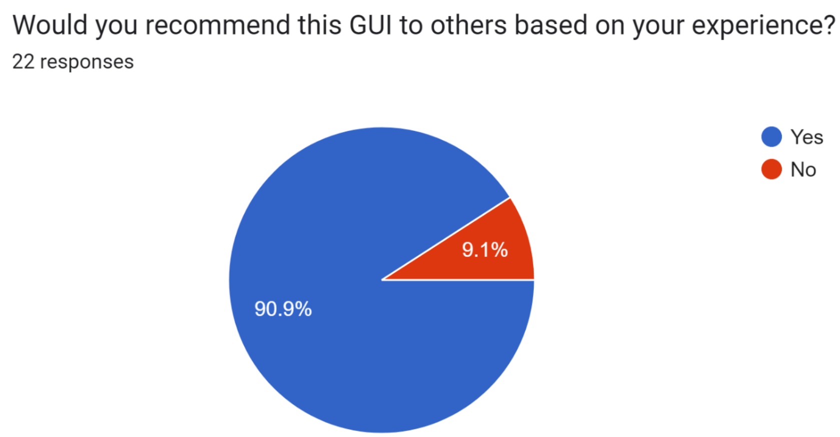


Figure 4.2.18: Majority would recommend the GUI to others based on their experience.

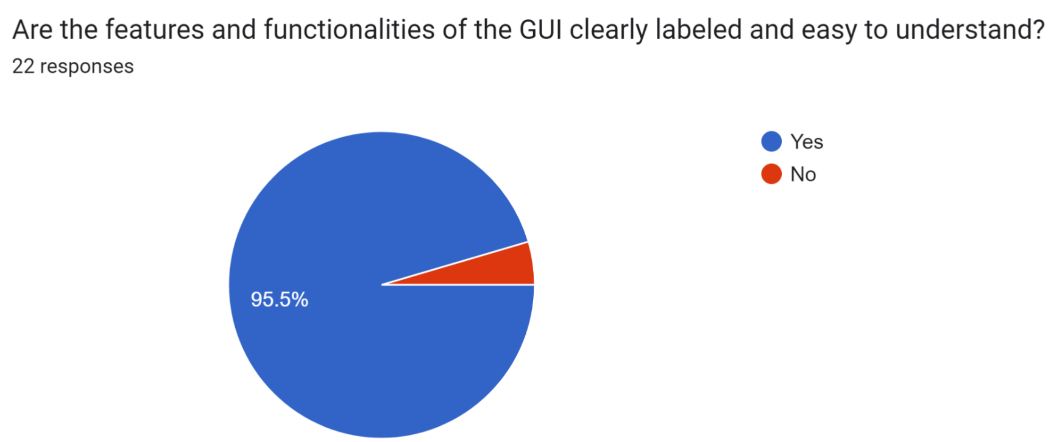


Figure 4.2.19: A large number of respondents feel that the features and functionalities of the GUI are clearly labelled and easy to understand.

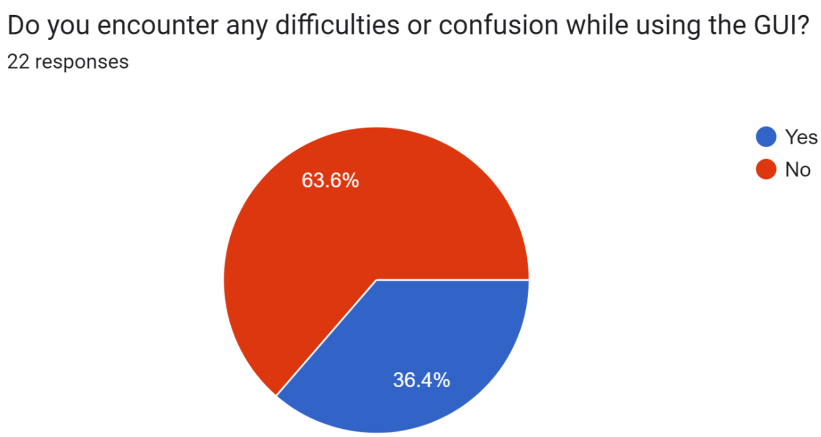


Figure 4.2.20: Only 36.4% of the respondents encountered some difficulties while using the GUI while the majority did not.

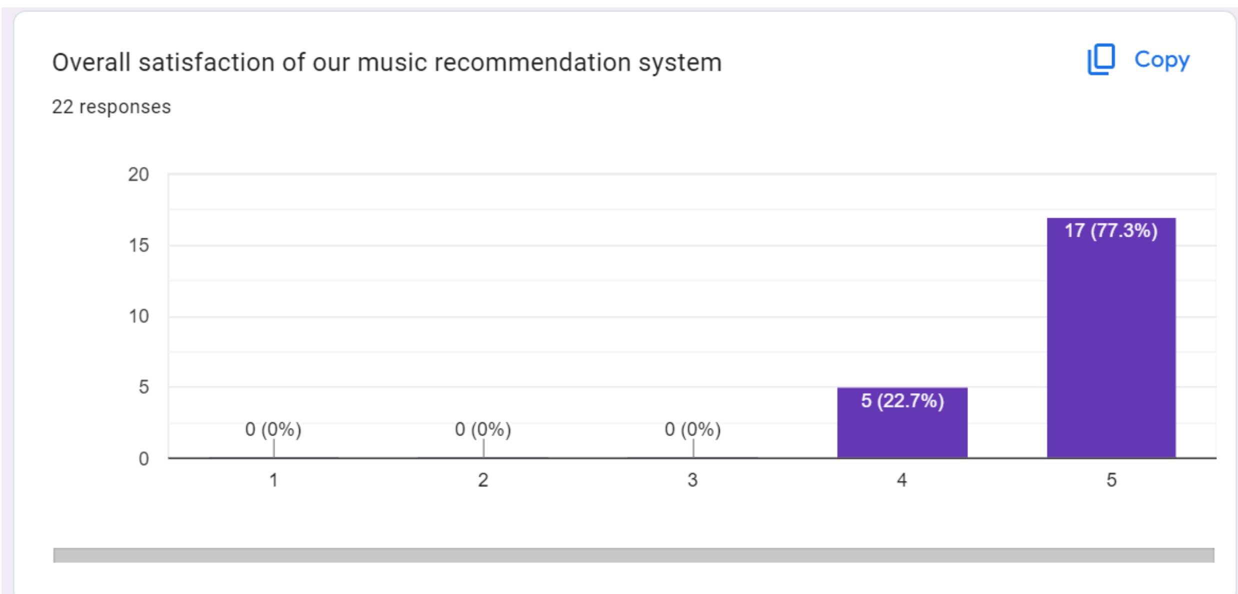


Figure 4.2.21: Percentage of overall satisfaction of our recommendation system

Regarding the Graphical user interface of our music recommendation system, we can see that most of the respondents think that the GUI is very appealing with easy to read font-style and font-size based on figure 4.2.2. In addition, the majority of the respondents rated 5 out of 5 for the overall of the GUI which makes it a great deal since it's related to one of the objectives to be achieved. In terms of efficiency and responsiveness, the statistics showed a positive result and most of the respondents think that they are likely to return to the GUI to use it again in the future, this might be due to the case that majorities feel that the GUI can adapt to their screen size without any issue and the navigation is very friendly. Another good thing is that most of them would consider recommending the system to others. From the responses, the recommender system seems to have the features and functionalities of the GUI clearly labelled and easy to understand because majority think of that. Just a slight number of respondents, 36.4% encountered difficulties or confusion while using the GUI. All in all, almost all of the respondents feel satisfied with our music recommendation system overall based on figure 4.2.3.

5. Discussion and Conclusion

5.1. Achievements

The project has successfully achieved its objectives, effectively fulfilling the aim of developing a robust song recommendation system capable of matching user predispositions with high accuracy and personalised recommendations. Firstly, the implementation of the GloVe algorithm, along with metrics like Cosine Similarity, Jaccard Similarity, and Pearson Correlation Coefficient, has enabled the system to recommend songs based on user input. This aspect aligns with the first objective, as the algorithmic approach allows for the generation of meaningful embeddings and the computation of similarity metrics crucial for recommendation accuracy.

Secondly, the improvement of an consumer- friendly Streamlined application has translated recommendation algorithms into an intuitive interface. This fulfilment immediately corresponds to the second one goal, as the application allows users to input preferences and explore recommended songs effortlessly. via incorporating consumer remarks and preferences into the advice method, the interface enhances the general consumer experience, thereby maximising application and impact.

Lastly, the project has successfully measured algorithm runtime to ensure computational efficiency for real-time recommendation generation, addressing the third objective. By quantifying the time required for processing song data and generating recommendations, the system ensures scalability and responsiveness, particularly crucial in the context of large-scale music libraries and user interactions. Overall, the project has achieved its objectives comprehensively, delivering a sophisticated song recommendation system that fulfils user needs effectively and efficiently.

5.2. Limitations and Future Works

Limitations

This recommendation system, constructed upon GloVe embeddings and similarity metrics, exhibits notable strengths in furnishing song recommendations grounded in semantic understanding. Nevertheless, a number of limitations hamper its efficacy. Primarily, while GloVe embeddings excel at capturing semantic meaning, they may struggle to grasp nuanced contexts or language intricacies inherent in song lyrics. This limitation can lead to inaccuracies in representing song meanings, potentially impacting the system's recommendation accuracy. Additionally, the system heavily relies on song lyrics as its primary data source, which may exclude instrumental or non-lyrical music from consideration, thereby limiting its scope and applicability.

Moreover, the system's effectiveness is contingent upon the availability and quality of song lyrics and GloVe embeddings. Sparse or incomplete song lyric data, coupled with gaps in GloVe coverage for certain words or phrases, can compromise the system's ability to generate accurate vector representations and compute meaningful similarity scores. Furthermore, the system's narrow focus on textual content overlooks other contextual factors influencing music preferences, such as genre, tempo, mood, or cultural context. Ignoring these nuances may result in recommendations that fail to fully align with users' preferences or expectations.

Another challenge lies in the selection of similarity metrics, such as cosine similarity, Jaccard similarity, and Pearson correlation coefficient. Different metrics prioritise distinct aspects of similarity, and their effectiveness may vary depending on the characteristics of the song dataset and user preferences. Additionally, the system may encounter the "cold start" problem, particularly for new users or songs with limited interaction data. Without sufficient user feedback or historical data, the system may struggle to provide accurate and personalised recommendations, potentially leading to suboptimal user experiences.

Addressing these limitations requires a holistic approach, encompassing refinements in text processing techniques, integration of additional contextual features beyond lyrics, enhancements in data coverage and quality, and the implementation of robust evaluation strategies to mitigate biases and enhance recommendation accuracy and relevance. By tackling these challenges, the recommendation system can elevate its ability to deliver meaningful and personalised song recommendations to users, enriching their music listening experiences.

Future works

First, the system's semantic understanding of song lyrics could be improved. In the future, more complex methods of encoding semantic meaning could be investigated, such as employing contextual embeddings or more advanced word embedding models. Furthermore, incorporating natural language processing (NLP) methods such as topic modelling or sentiment analysis could increase recommendation accuracy by offering a more in-depth understanding of lyrical content.

Secondly, by combining data from several sources, the system may develop into a multimodal one. A more thorough understanding of each song could be achieved by incorporating metadata such as genre, artist information, and release date with audio characteristics like melody, rhythm, and timbre. The system could more accurately customise recommendations to match user preferences if it included a greater variety of modalities.

In addition, context awareness and personalization may be the main topics of future work. More tailored and context-aware recommendations may be possible with developments in machine learning and recommendation systems. By optimising recommendations in real-time based on changing user interactions and environmental cues, techniques like contextual bandits and reinforcement learning have the potential to increase user satisfaction.

Furthermore, reliable assessment techniques and reference datasets are essential for precisely evaluating the performance of recommendation systems. Subsequent investigations may aid in the creation of benchmark datasets and standardised assessment procedures that are especially suited to music recommendation tasks. As a result, it would be easier to compare various algorithms fairly and find viable ideas for additional research.

Additionally, it is crucial to guarantee that the recommendations are transparent and comprehensible. In order to increase trust and understanding, future work could concentrate on creating methods for users to understand and be able to defend recommendation outcomes. Transparency and user satisfaction could be increased by incorporating interpretable models offering user-friendly explanations and allowing user feedback on recommendations.

It is imperative to prioritise addressing ethical considerations such as privacy, diversity, and justice. Subsequent studies may examine methods for reducing biases, encouraging diversity in suggestions, and protecting user privacy. Recommendation systems could be deployed responsibly and ethically if privacy-preserving measures, diversity constraints, and algorithms with consideration for fairness were implemented. The recommendation system may develop to better suit the various needs and preferences of users while respecting the values of justice, transparency, and ethical use by following these directions for future research and embracing ongoing innovation and collaboration within the research community.

Reference & Source

1. Velankar, M., & Kulkarni, P. (2023). Music Recommendation Systems: Overview and Challenges. Advances in Speech and Music Technology: Computational Aspects and Applications, 51. In *Google Books*. Springer Nature. https://books.google.com.my/books?hl=en&lr=&id=cdWkEAAAQBAJ&oi=fnd&pg=PA51&ots=7Q87ek_aAv&sig=PpJJx5-IHXv2NAiSP_-J21-ZJmk&redir_esc=y#v=onepage&q&f=false

2. Kathavate, S. (2021, February 2). *Music Recommendation System using Content and Collaborative Filtering Methods* . International Journal of Engineering Research & Technology (IJERT). https://d1wqtxts1xzle7.cloudfront.net/66162758/music_recommendation_system_using_IJERTV10IS020071-libre.pdf?1617347896=&response-content-disposition=inline%3B+filename%3DIJERT_Music_Recommendation_System_using.pdf&Expires=1714549205&Signature=CbC~65iiJ2s4gvCGw7LtCclfZsqX8OoA-b4yPoctdmFIyJI2kGZfJBHWf2ZvmIl~CSfnnikngqDwsNSGnhNMXbIiTCm97GDzTEolgnXv6GkrKve35Sv41AgazlDFG7PemWwG8C7BT9EgwNkb6VpfcH0YTOo~wRBQCUTJVMawb8nEFF2J3JNQOwryvxXiLqzHYeWksdZaLF8BTPN0kxFqjtnhLs004aVUTD8~0AQDIrXjnS2ZwWqORvuSs3Zg11RLAWGXGMkFVQ-mf3JISkY2hx6hN4uLppvoQ3B9~8E1EOmUfi5DWMG3gDBml~LhQpW~1zo30KoxG9PbDlqULc6-qA__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

3. Feng, C., Liang, J., Song, P., & Wang, Z. (2020). A fusion collaborative filtering method for sparse data in recommender systems. *Information Sciences*, *521*, 365–379. https://doi.org/10.1016/j.ins.2020.02.052

4. Schedl, M. (2019). Deep Learning in Music Recommendation Systems. *Frontiers in Applied Mathematics and Statistics*, *5*. https://doi.org/10.3389/fams.2019.00044

5. None Swarnima, & Mala Saraswat. (2024). *Music Recommendation System Using Deep Learning and Machine Learning*. https://doi.org/10.1109/ic2pct60090.2024.10486423

6. Deldjoo, Y., Schedl, M., Cremonesi, P., & Pasi, G. (2020). Recommender Systems Leveraging Multimedia Content. *ACM Computing Surveys*, *53*(5), 1–38. https://doi.org/10.1145/3407190

7. Sheikh Fathollahi, M., & Razzazi, F. (2021). Music similarity measurement and recommendation system using convolutional neural networks. *International Journal of Multimedia Information Retrieval*, *10*(1), 43–53. https://doi.org/10.1007/s13735-021-00206-5

8. Zhang, Y. (2022). Music Recommendation System and Recommendation Model Based on Convolutional Neural Network. *Mobile Information Systems*, *2022*, 1–14. https://doi.org/10.1155/2022/3387598

9. Mukhopadhyay, S., Kumar, A., Parashar, D., & Singh, M. (2024). Enhanced Music Recommendation Systems: A Comparative Study of Content-Based Filtering and K-Means Clustering Approaches. *Revue d'Intelligence Artificielle*, *34*(1), 365–376. https://doi.org/10.18280/ria.340138

10. Wahyudi, K., Latupapua, J., Chandra, R., & Girsang, A. S. (2020). Hotel Content-Based Recommendation System. *Journal of Physics: Conference Series*, *1485*, 012017. https://doi.org/10.1088/1742-6596/1485/1/012017

11. Niyazov, A., Mikhailova, E., & Egorova, O. (2021, May 1). *Content-based Music Recommendation System*. IEEE Xplore. https://doi.org/10.23919/FRUCT52173.2021.9435533

12. Pulis, M., & Bajada, J. (2021). Siamese Neural Networks for Content-based Cold-Start Music Recommendation. *Conference on Recommender Systems*. https://doi.org/10.1145/3460231.3478847

13. Katkam, S., Atikam, A., Mahesh, P., Chatre, M., Kumar, S. S., & R, S. G. (2023, May 1). *Content-based Movie Recommendation System and Sentimental analysis using ML*. IEEE Xplore. https://doi.org/10.1109/ICICCS56967.2023.10142424

14. Christina, N., Febrian Sanjaya, Valentina Tiara Cahyaningtyas, Ivan Sebastian Edbert, & Derwin Suhartono. (2023). *Distance Metric Analysis in Recommendation System Using Content-Based Filtering Method*. https://doi.org/10.1109/icoiact59844.2023.10455814

15. Sakti, S. M., Laksito, A. D., Sari, B. W., & Prabowo, D. (2022). Music Recommendation System Using Content-based Filtering Method with Euclidean Distance Algorithm. *2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*. https://doi.org/10.1109/icitisee57756.2022.10057753

16. Mao, Y., Zhong, G., Wang, H., & Huang, K. (2022). Music-CRN: an Efficient Content-Based Music Classification and Recommendation Network. *Cognitive Computation*. https://doi.org/10.1007/s12559-022-10039-x