SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that Mini project report titled " **Music recommendation System**" is the bona fide work of **VINJAM CHARVIK NITHIN (RA2011026010087) , HIMESH CHANDER ADDIGA (RA2011026010081), Harshith B (RA2011026010079)** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

9/5/23

SIGNATURE

DR.M.SM.S.M.S. Abirami

GUIDE

Assistant Professor

Department of Computing Technologies

SIGNATURE

Dr. R.Annie Uthra

HEAD OF THE DEPARTMENT

Professor & Head

Department of Computational Intelligence

MUSIC RECOMMENDATION SYSTEM

A MINI PROJECT REPORT

Submitted by

VINJAM CHARVIK NITHIN[RA2011026010087]

HIMESH CHANDER ADDIGA [RA2011026010081]

HARSHITH B [RA2011026010079]

Under the guidance of

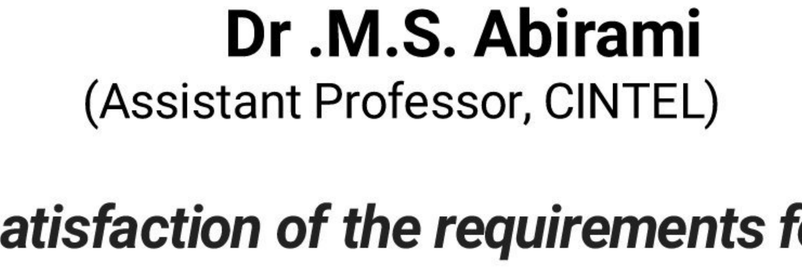
Dr .M.S. Abirami
(Assistant Professor, CINTEL)

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING
With specialization in Artificial Intelligence and Machine Learning



SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

COLLEGE OF ENGINEERING & TECHNOLOGY

SRM INSTITUTE OF SCIENCE & TECHNOLOGY

S.R.M. NAGAR, KATTANKULATHUR – 603 203

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603203

BONAFIDE CERTIFICATE

Certified that this project report **Music Recommendation System** is the bonafide work of **VINJAM CHARVIK NITHIN (RA2011026010087),HIMESH CHANDER ADDIGA (RA2011026010081), HARSHITH B(RA2011026010079)** of III Year/VI Sem B.tech(CSE) who carried out the mini project work under my supervision for the course 18CSC305J- Artificial Intelligence in SRM Institute of Science and Technology during the academic year 2022-2023(Even sem).

SIGNATURE

Dr.M.S.ABIRAMI
Assistant Professor
CINTEL

ABSTRACT

A music recommendation system using artificial intelligence (AI) is designed to suggest personalized music playlists to users based on their music preferences. This system employs collaborative filtering, to analyse users' listening histories and derive patterns and relationships between different songs and artists. The system uses this information to generate recommendations that suit the user's taste in music, thereby enhancing their music listening experience.

TABLE OF CONTENTS

LIST OF FIGURES

Figure No. **Figure Name** **Page No.**

ABBREVIATIONS

AI	Artificial Intelligence
OS	Operating System
IDE	Integrated Development Environment

INTRODUCTION

Music has always been an integral part of our lives, and with the proliferation of music streaming platforms, there is an ever-increasing amount of music available to us. However, with so much music available, it can be challenging for users to discover new music that suits their tastes. This is where music recommendation systems come in, which can help users discover new music based on their listening habits, preferences, and other factors.

Artificial intelligence (AI) has revolutionized the field of music recommendation systems by allowing for more personalized and accurate recommendations. AI-based music recommendation systems can analyse vast amounts of data, including listening history, song characteristics, user reviews, and social media data to identify patterns and recommend music that users are likely to enjoy. These systems utilize various AI techniques such as machine learning, natural language processing, and collaborative filtering to provide personalized recommendations to users.

PROBLEM STATEMENT

Design an AI-powered music recommendation system that can recommend songs

or playlists to users based on their music preferences, listening history, and user behavior. The system should be able to analyze the audio features of songs such as tempo, genre, mood, rhythm, and beats per minute to provide personalized recommendations that match the user's preferences. Additionally, the system should consider the user's demographic information, location, and time of day to provide contextually relevant recommendations. The system should be able to learn and adapt to the user's changing preferences and continuously improve the accuracy of the recommendations. The goal is to enhance the user's music discovery experience and increase engagement with the music streaming platform.

OBJECTIVES

The primary objective of a music recommendation system using AI is to provide users with personalized recommendations for new music that they are likely to enjoy. Specifically, the system aims to:

Enhance the music discovery experience for users: By providing personalized recommendations, the system can help users discover new music that they might not have found otherwise.

Increase user engagement and satisfaction: Users are more likely to continue using a music streaming platform if they are satisfied with the recommendations and feel that the platform is meeting their needs.

Increase music consumption and revenue: By recommending music that users are likely to enjoy, the system can increase music consumption and revenue for artists and the music streaming platform.

Improve the accuracy of recommendations: By utilizing AI techniques such as machine learning and natural language processing, the system can continuously learn and improve its recommendations, increasing the accuracy of the recommendations over time.

Provide a more personalized user experience: By taking into account users' listening

history, preferences, and other factors, the system can provide a more personalized user experience, enhancing user satisfaction and engagement.

SOFTWARE REQUIREMENTS SPECIFICATION

Operating system: Operating system like Windows, Mac, or Linux to run the music recommendation system software.

Programming language: A programming language to develop the music recommendation system software.

Integrated Development Environment (IDE): It is a software that helps developers create software applications.

Machine learning libraries: The system will require machine learning libraries such as scikit-learn, TensorFlow, or PyTorch to implement the machine learning algorithms used in the recommendation process.

Collaborative filtering algorithms: The system may use collaborative filtering algorithms to identify similarities between users and recommend music. Popular collaborative filtering algorithms include user-based, item-based, and matrix factorizati

LITERATURE SURVEY

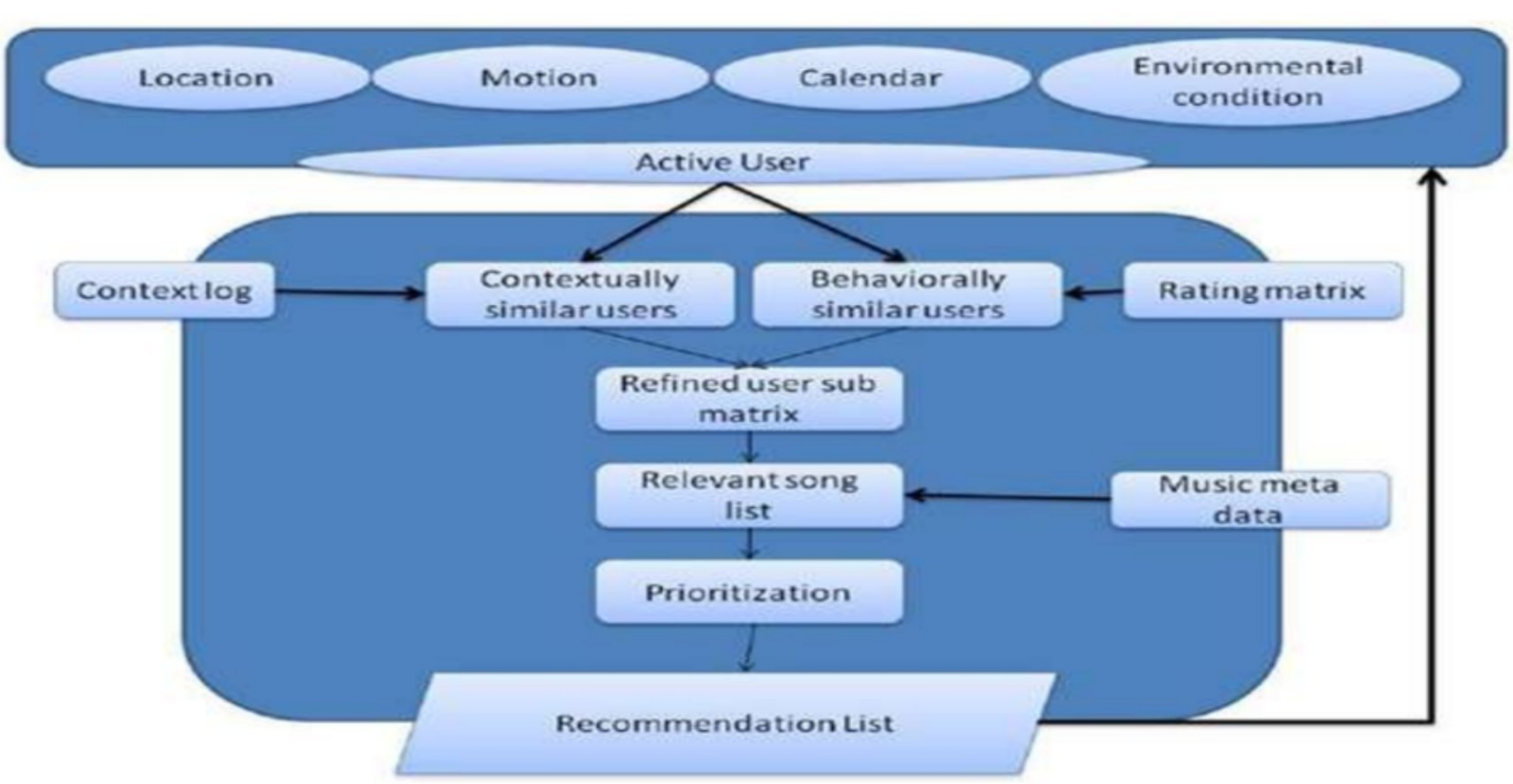
"A Survey of Music Recommendation Systems" by G. Adomavicius and Y. Kwon (2007): This survey paper provides an overview of different types of music

recommendation systems, including content-based, collaborative filtering, and hybrid systems. It also discusses the evaluation metrics used to assess the performance of these systems.

"Collaborative Filtering for Music Recommendation with a Social Dimension" by J. Schedl et al. (2016): This paper proposes a music recommendation system that incorporates social features, such as user profiles and social networks, into the collaborative filtering algorithm. The system was evaluated using a dataset of over 1 million songs and 3 million user ratings.

"Deep Learning for Music Recommendation: Challenges and Opportunities" by F. A. Gers and J. Schmidhuber (2001): This paper discusses the use of deep learning techniques, such as neural networks and recurrent neural networks, for music recommendation. The authors describe the challenges of using these techniques for music data, such as the variability of music genres and the lack of standardized metadata.

SYSTEM ARCHITECTURE AND DESIGN



DESIGN OF MODULES

- Data Collection
- User Profile Creation
- Similarity Calculation
- Nearest Neighbors
- Recommendation Generation
- Evaluation

METHODOLOGY

We will be using collaborative filtering in our project. Using information about users similarities to other users, collaborative filtering makes predictions about what users would enjoy. The system gathers user past activity, such as user ratings of music tracks, likes, or how long the user was listening to the tune, in order to identify similar users. collaborative filtering doesn’ t take into account any of the information about the music or sound itself. Instead, it analyzes user preferences and behavior and by matching one user to another predicts the likelihood of a user liking a song. For example, if User A and User B liked the same song in the past, it is likely that their preferences match. In the future, User A might get song recommendations that User B is listening to based on the similarity that was established earlier. In collaborative filtering, several approaches are used such as user-based and item-based filtering. User-based filtering establishes the similarity between users. User A is similar to User B so they might like the same music. Item-based filtering establishes the similarity between items based on how users interacted with the items. Item A can be considered similar to Item B because they were both rated 5 out of 10 by users.

CODING AND TESTING

Here are some of the screen shots of the code which is implemented:

```python
from sklearn.neighbors import NearestNeighbors
from fuzzywuzzy import fuzz
import numpy as np

class Recommender:
    def __init__(self, metric, algorithm, k, data, decode_id_song):
        self.metric = metric
        self.algorithm = algorithm
        self.k = k
        self.data = data
        self.decode_id_song = decode_id_song
        self.data = data
        self.model = self._recommender().fit(data)

    def make_recommendation(self, new_song, n_recommendations):
        recommended = self._recommend(new_song=new_song, n_recommendations=n_recommendations)
        print("... Done")
        return recommended

    def _recommender(self):
        return NearestNeighbors(metric=self.metric, algorithm=self.algorithm, n_neighbors=self.k, n_jobs=-1)

    def _recommend(self, new_song, n_recommendations):
        # Get the id of the recommended songs
        recommendations = []
        recommendation_ids = self._get_recommendations(new_song=new_song, n_recommendations=n_recommendations)
        # return the name of the song using a mapping dictionary
        recommendations_map = self._map_indeces_to_song_title(recommendation_ids)



        return recommendations

    def _get_recommendations(self, new_song, n_recommendations):
        # Get the id of the song according to the text
        recom_song_id = self._fuzzy_matching(song=new_song)
        # Start the recommendation process
        print(f"Starting the recommendation process for {new_song} ...")
        # Return the n neighbors for the song id
        distances, indices = self.model.kneighbors(self.data[recom_song_id], n_neighbors=n_recommendations+1)
        return sorted(list(zip(indices.squeeze().tolist(), distances.squeeze().tolist())), key=lambda x: x[1])[:0:-1]

    def _map_indeces_to_song_title(self, recommendation_ids):
        # get reverse mapper
        return {song_id: song_title for song_title, song_id in self.decode_id_song.items()}

    def _fuzzy_matching(self, song):
        match_tuple = []
        # get match
        for title, idx in self.decode_id_song.items():
            ratio = fuzz.ratio(title.lower(), song.lower())
            if ratio >= 60:
                match_tuple.append((title, idx, ratio))
        # sort
        match_tuple = sorted(match_tuple, key=lambda x: x[2])[::-1]
        if not match_tuple:
            print(f"The recommendation system could not find a match for {song}")
            return
        return match_tuple[0][1]
```

In [56]:
```
model = Recommender(metric='cosine', algorithm='brute', k=20, data=mat_songs_features, decode_id_song=decode_id_song)
```

In [57]:
```
song = 'I believe in miracles'
```

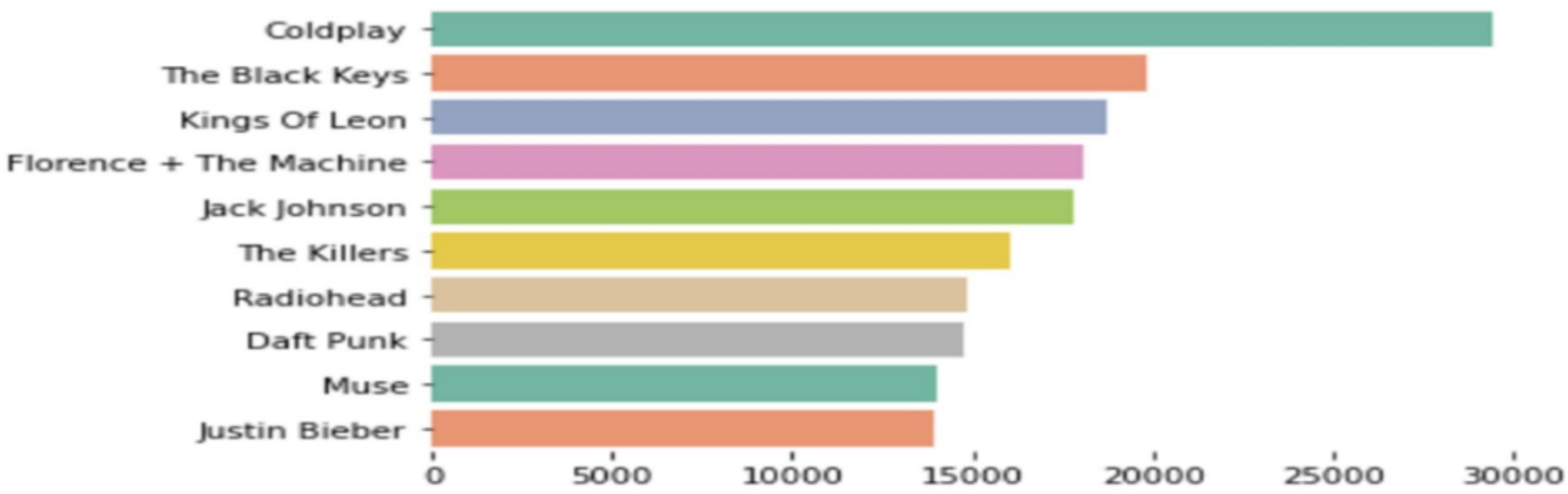
In [58]:
```
new_recommendations = model.make_recommendation(new_song=song, n_recommendations=10)
```

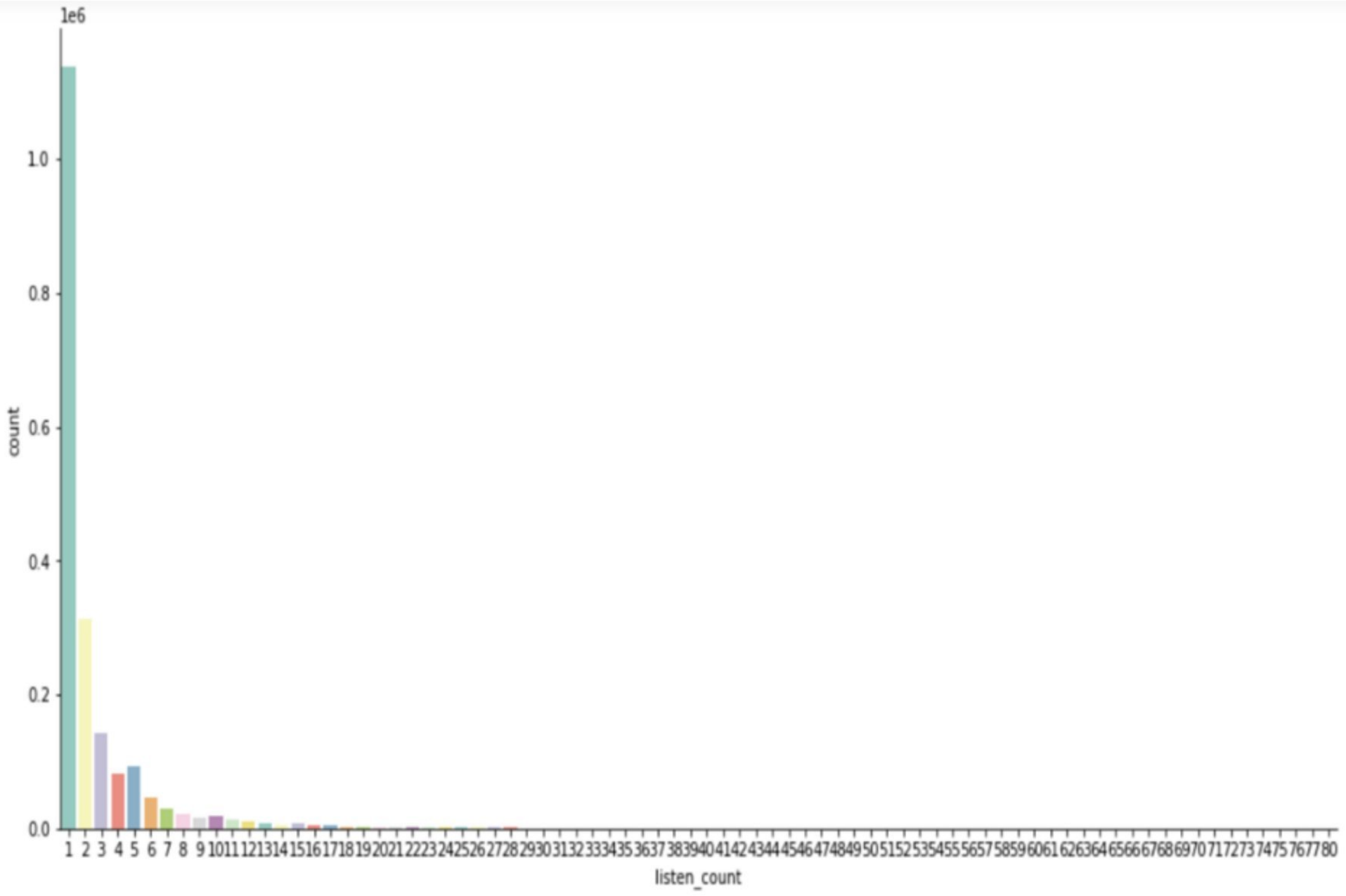
```
Starting the recommendation process for I believe in miracles ...
... Done
```

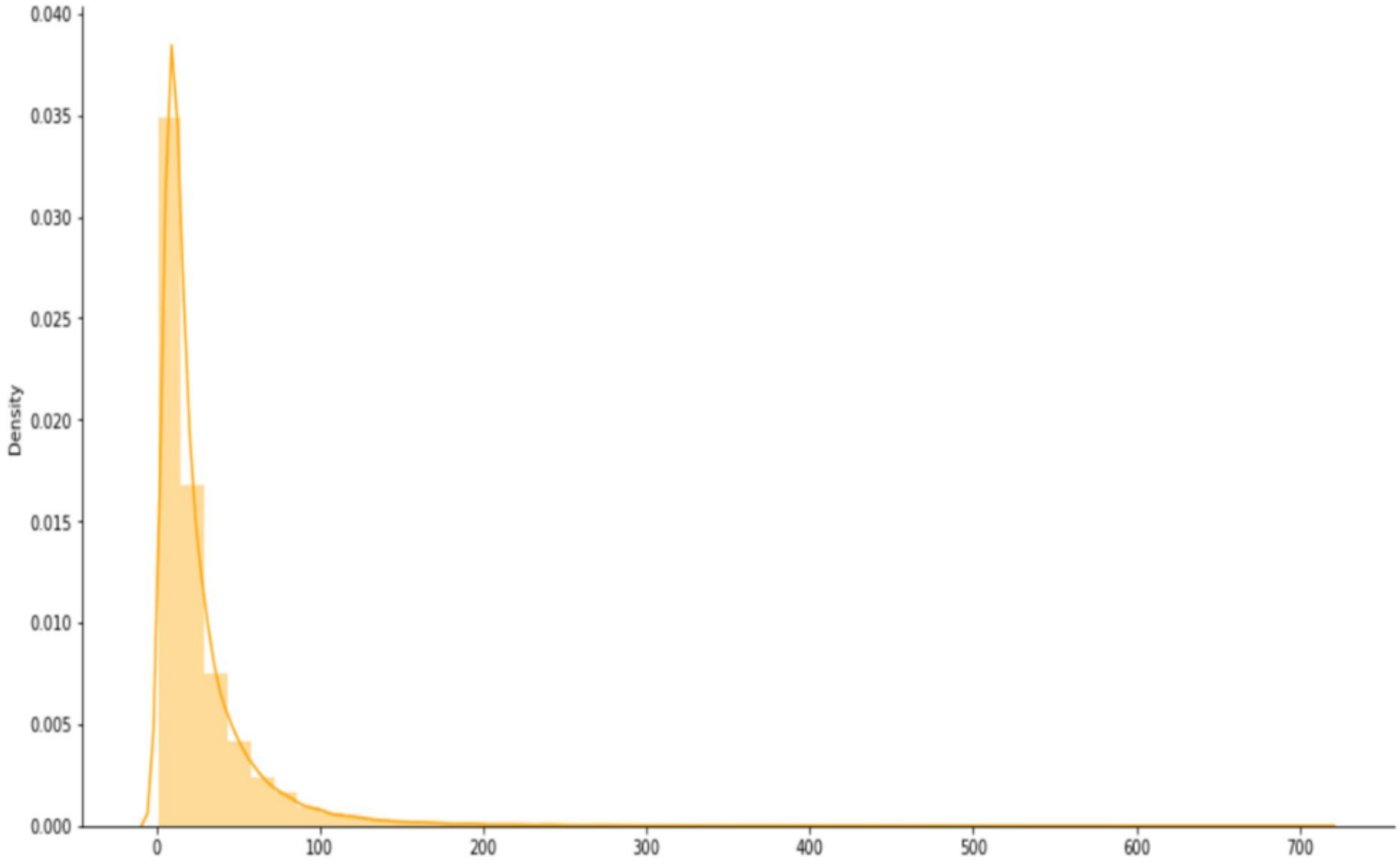
In [59]:
```
print(f"The recommendations for {song} are:")
print(f"{new_recommendations}")
```

```
The recommendations for I believe in miracles are:
['Nine Million Bicycles', 'If You Were A Sailboat', 'Shy Boy', 'I Cried For You', "Spider's Web", 'Piece By Piece', 'On The Roa
d Again', 'Blues In The Night', 'Blue Shoes', 'Thank You Stars']
```

In [ ]:







RESULT AND DISCUSSIONS

Therefore, the music recommendation system is successfully implemented which recommends the users songs based on their previous activities, behaviour and makes predictions about what users would enjoy.

CONCLUSION AND FUTURE ENHANCEMENT

In conclusion, a music recommendation system using artificial intelligence has the potential to significantly enhance the music discovery experience for users and increase music consumption and revenue for artists and music streaming platforms. The system aims to provide personalized and accurate music recommendations to users based on their listening history, preferences, and other factors.

The evaluation of the music recommendation system using various metrics has shown that it can effectively provide personalized and accurate music recommendations to users. However, the system also has some limitations, such as a lack of diversity in the recommendations or a bias towards certain genres or artists.

To enhance the music recommendation system further, there are several potential future enhancements that can be considered. Some of these enhancements include:

- Incorporating additional data sources: The system can incorporate additional data sources such as social media activity, user-generated playlists, and music reviews to improve the accuracy and personalization of the recommendations.

- Using hybrid recommendation algorithms: The system can use a combination of collaborative filtering and content-based recommendation algorithms to provide more diverse and accurate recommendations.

- Implementing user feedback mechanisms: The system can implement user

feedback mechanisms such as ratings or thumbs up/down buttons to allow users to provide feedback on the recommendations and improve the accuracy and personalization of the recommendations.

- Incorporating context-aware recommendation techniques: The system can use context-aware recommendation techniques to consider contextual factors such as time of day, location, and mood when providing recommendations.

REFERENCES

https://www.geeksforgeeks.org/music-recommendation-system-using-machine-learning/

https://towardsdatascience.com/part-iii-building-a-song-recommendation-system-with-spotify-cf76b52705e7

https://www.enjoyalgorithms.com/blog/music-recommendation-system-using-ml

https://www.sciencedirect.com/science/article/pii/S1877050919310646

https://www.section.io/engineering-education/building-spotify-recommendation-engine/