Music Recommendation System Using Natural Language Processing

By

Chege Caroline Njeri

Student number: 082908

Master of Science in Data Science and Analytics

2024

Music Recommendation System Using Natural Language Processing

By

Chege Caroline Njeri

Student number: 082908

Submitted in Partial fulfilment of the Requirements for the Degree of Master of Science in Data Science and Analytics at Strathmore University

Institute of Mathematical Sciences

Strathmore University

Nairobi, Kenya

June 2024

Declaration and Approval

Declaration

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Student Name: Caroline Chege

Sign: _____ _____ Date: ___9th April 2024_____

Approval

The thesis of Caroline Chege was reviewed and approved for examination by the following:

Dr Kennedy Senagi,

Institute of Mathematical Sciences

Strathmore University

Sign: _____ Date: 11th April 2024

Dr. Godfrey Madigu,

Dean, Institute of Mathematical Sciences,

Strathmore University

Dr. Bernard Shibwabo,

Director of Graduate Studies,

Strathmore University

Abstract

Music recommendation systems have become increasingly popular in recent years, facilitating personalized music discovery for users worldwide. This dissertation explores the application of natural language processing (NLP) and machine learning techniques in developing a music recommendation system. The study involves building a collection of music lyrics databases, analyzing the lyrics using NLP methods (such as TF-IDF and similarity/distance metrics), and integrating these findings into a recommendation model. The cosine similarity model was evaluated and recorded an accuracy of 96%, precision of 95%, recall of 96% and F1-score of 95%. Therefore, incorporating lyrics-based features in music recommendation systems can improve user experience in consuming recommendations of similar and relevant music.

Table of Contents

Declaration and Approval ii

Abstract iii

Table of Contents iv

List of Figures vi

List of Tables vii

List of Abbreviations viii

Acknowledgements ix

Chapter 1: Introduction 1

1.1 Background 1

1.2 Problem Statement and Justification 2

1.3 Research Objectives 3

1.4 Research Questions 3

1.5 Assumptions and Scope 4

Chapter 2: Literature Review 5

2.1 Big Data and its Impact in Various Industries 5

2.2 Natural Language Processing 5

2.3 Music Recommendation Systems 8

Chapter 3: Methodology 11

3.1 Introduction 11

3.2 Data Source and Dataset 12

3.3 Data Pre-Processing 14

3.3.1 Text Preprocessing 15

3.3.2 Feature Engineering 16

3.4 Text Vectorization 18

3.5 Building the Recommender System 20
3.5.1 Cosine Similarity 20
3.5.2 Levenshtein Distance 21
3.5.3 Ranking Music 22
3.6 Performance Evaluation 22
3.7 Deployment 24
Chapter 4: System Design and Architecture 25
4.1 Workflow 25
Chapter 5: System Implementation and Testing 27
5.1 System Implementation 27
5.1.1 Frontend 27
5.1.2 Backend 27
Chapter 6: Discussion of Results 30
6.1 Exploratory Data Analysis 30
6.1.1 Univariate Data Analysis 30
6.1.2 Bivariate Data Analysis 32
6.1.3 Word Cloud 34
6.2 Music Recommendation System Performance Results 36
Chapter 7: Conclusions, Recommendations and Future works 39
7.1 Conclusions of the Study 39
7.2 Future Works 39
References 40
Appendices 44
Appendix A: Similarity Report 44
Appendix B: Ethical Clearance Release Letter 45

List of Figures

Figure 2.1: Recent developments in NLP 7
Figure 3.1: Crisp-DM methodology 11
Figure 3.2: Sample row of the dataset and respective features 13
Figure 3.3: Data Preprocessing 14
Figure 3.4: Root Word 15
Figure 3.5: Word Frequency 17
Figure 3.6: Sentiment analysis 18
Figure 3.7: TF-IDF Vectorizer 19
Figure 4.1: System architecture and design 25
Figure 4.2: Use case diagram 26
Figure 5.1: Song selection 28
Figure 5.2: Streamlit recommendation output sample 28
Figure 5.3: Back-end implementation of Streamlit application 29
Figure 6.1: Word Counts 30
Figure 6.2: Word Counts distribution 31
Figure 6.3: Word Length 31
Figure 6.4: Sentiment Scores Distribution 32
Figure 6.5: Bivariate data analysis 33
Figure 6.6: Sentiment trends over time for each genres 34
Figure 6.7: Word Cloud 35
Figure 6.8: Cosine Similarity Matrix 36
Figure 6.9: Recommendation output sample 36
Figure 6.10: Confusion Matrix 37

List of Tables

Table 3.1: Data attributes and description 13

List of Abbreviations

NPL	Natural Language Processing
TF-IDF	Term Frequency- Inverse Document Frequency
IR	Information Retrieval
CRISP-DM	Cross Industry Standard Process for Data Mining
API	Application Programming Interface
URL	Uniform Resource Locator

Acknowledgements

I am immensely grateful to God, the Almighty, for granting me the strength, wisdom, and perseverance to complete this thesis.

Secondly, I extend my deepest gratitude to my supervisor Dr Kennedy Senagi, for sharing his expertise, his unwavering support, guidance, and encouragement. His mentorship has been invaluable in shaping the course of this research.

I am thankful to Strathmore University administrative staff and the office of graduate studies for providing the necessary support, resources, and facilities for this study.

My heartfelt appreciation goes to my family for their unconditional love, unwavering faith, and endless support. Their encouragement has been a driving force behind my success.

I am also grateful to my friends and colleagues for their insightful discussions, and words of encouragement.

I would also like to express my gratitude to all the participants who generously contributed their time and insights to this research.

Chapter 1: Introduction

1.1 Background

In the human context, music can be defined as sounds crafted by individuals that elicit enjoyable auditory experiences. It is a product of human thought and engagement, seemingly intertwined with nature and having an existence beyond humans (Lawendowski & Bieleninik, 2017, 85-99). It serves as one of the most universal means of expression and communication across diverse cultures and age groups worldwide. Anthropological and ethnomusicological research indicates that music has been an integral aspect of the human experience for thousands of years (Welch et al., 2020).

Music is found universally among human cultures, and it serves as a common source of emotional and pleasurable experiences that evokes both physical and emotional responses. The examination of how the brain processes music, particularly in terms of perceiving melody, harmony and rhythm has traditionally been centered around the auditory aspects. (Vuust et al., 2022, 287-305). One notable characteristic of music lies in its capacity to communicate emotions. It functions as a global language which enables the expression and comprehension of sentiments that might be challenging to articulate. For instance, a melancholic melody has the power to evoke tears and offer comfort during times of grief and sorrow. Similarly, a lively rhythm has the ability to spark a surge of vitality and elation (Alawi, 2023).

As we engage with music, there's a set of intricate neural activities that unfold within our brains. Diverse regions of the brain collaborate to process various aspects of the musical encounter or experience including pitch, rhythm, melody and lyrics. (Alawi, 2023) The literature specifically by (Schäfer et al., 2013), extensively details the various functions, roles, and psychological applications of music. Their investigation observed that multiple empirical studies suggested the categorizing of musical functions based on four dimensions: cognitive, emotional, social/cultural and psychological/arousal functions.

Furthermore, music has distinctly been used as a therapeutic and supportive instrument. The formal inception of professional music therapy (MT) took place during World War II, where it functioned as a form of musical anesthesia during surgery and a psychiatric approach for treating war veterans.

(Lawendowski & Bieleninik, 2017, 85-99). Through well-structured music-based interventions, individuals can enhance their intellectual capabilities and cultivate a sense of personal identity by discarding negative self-perceptions, leading to potential improvements in self-esteem. (Lawendowski & Bieleninik, 2017, 85-99). While music is inherently pleasurable, its impact extends beyond mere entertainment (Welch et al., 2020).

The Music industry stands out as a sector characterized by consistent growth and plays a significant role in generating substantial profits (Greenberg & Rentfrow, 2017, 50-56). Today, music is present in almost every aspect of our lives and is disseminated through diverse channels. A few decades ago, radio stood out as the predominant mode of music consumption. However, in the contemporary landscape, the music industry has undergone a remarkable transformation propelled by digital distribution services and real-time streaming technologies (Lekamge et al., 2017, 205-211).

In 2019, the international music industry produced $20.2 billion according to the International Federation of the Phonographic Industry's (IFPI) annual report. In 2020, music streaming comprised 62.1% of the total revenue for the global music industry (Zehr, 2021). In 2022, revenue from music streaming applications reached $43.3 billion, marking a 15% rise from the previous year. Nearly 50% of this revenue originated from the United States. Spotify holds the lead as the most subscribed music streaming service. However, when considering overall usage, Youtube stands significantly ahead with 2.5 billion users (Curry, 2024).

In this proposal, we seek to explore the use of NLP in music recommendation systems and discuss the potential benefits and challenges of using this technology. We will also present case studies and examples of existing systems that use NLP to recommend music based on lyrics, and consider their effectiveness and limitations.

1.2Problem Statement and Justification

Music recommendation systems that use natural language processing (NLP) to analyze the lyrics of songs have the potential to provide more personalized and nuanced recommendations compared to traditional approaches that rely on metadata such as artist, genre, and popularity. However, there are several challenges and limitations to using NLP for music recommendation, including the

complexity and variability of natural language, the subjectivity of music preferences and sentiments, and the need for large amounts of data and computational resources.

The availability of large-scale lyric datasets and advancements in NLP models offer an opportunity to explore the incorporation of semantic understanding of lyrics into music recommendation systems. The challenge, however, lies in developing NLP-based approaches that not only extract meaningful information from lyrics, but also model user preferences effectively. This study seeks to address this challenge by investigating the development of NLP-driven music recommendation systems that consider both lyrics and user behavior.

1.3 Research Objectives

Overall, the goal is to provide a comprehensive overview of NLP-based music recommendation systems and their role in the music industry. We will explore how NLP can be used to recommend music based on the lyrics and how this technology can help music lovers discover new songs that match their preferences and moods.

The specific research objectives will be to:

a) Investigate natural language processing and other machine learning techniques used in music recommendation systems.

b) Build a collection of databases of music lyrics and their respective genre.

c) Develop a music recommendation system using NLP approaches.

d) To deploy the model on a web platform.

1.4 Research Questions

a) What are some of the natural language processing techniques applicable to analyzing music lyrics, and how do they contribute to the effectiveness of a music recommendation system

b) How can a diverse collection of music lyrics along with other data be efficiently compiled

c) How can methods be optimally integrated into a music recommendation system to enhance the overall recommendation accuracy.

d) What are the best practices for deploying recommendation models that combine lyric-based features.

1.5 Assumptions and Scope

a) It is assumed that the use of NLP in music recommendation can provide more personalized and relevant recommendations compared to traditional approaches that rely on metadata.

b) It is assumed that NLP algorithms can accurately analyze the meaning and sentiment of lyrics and identify similar songs based on these factors.

c) It is assumed that there is enough annotated lyrics data available for training and evaluating NLP models for music recommendation.

For this dissertation, we shall only be considering songs whose lyrics are in English.

Chapter 2: Literature Review

2.1 Big Data and its Impact in Various Industries

The availability of supercomputing capabilities and increased storage capacities has allowed the development of new technologies for data storing, computing, and analyzing. These developments have led to the existence of what is now known as Big data (Hujran et al., 2020). The term became widespread around 2011. According to (Hujran et al., 2020) the current hype can be attributed to the promotional initiatives by IBM and other leading technology companies who invested in building the niche analytics market.

The digitization of the world led to the emergence of some of the most important companies in the world today like Apple, Google, Facebook, Twitter. However, there are a number of traditional businesses and industries that failed to seize such opportunities to adapt to the evolving technologies to survive the market (Hujran et al., 2020).

In recent years, Big data has been a significant consideration and addition to many businesses, by creating value. However, to create value, strong analytics capabilities are required. Big data creates value by solving various problems and addressing different business challenges (Turner et al., 2013). These advantages have been felt in multiple industries, and the entertainment industry, especially music, has not been left behind. According to (Inellipaat, 2022), people having access to digital gadgets has led to the generation of large amounts of data.

The benefits of this wave have been greatly felt by this industry and include but not limited to; predicting the interest of audiences, optimization or on-demand scheduling of media streams, getting insights from customer reviews and effective targeting of advertisements. (Inellipaat, 2022)

2.2 Natural Language Processing

Natural language processing is an area of research based on linguistics, computer science and artificial intelligence that is concerned with the interaction between computers and human language (Chowdhruy & Gobinda, 2003).

According to (Khurana et al., 2023, 3713-3744), Natural Language Processing (NLP) lies at the intersection of Artificial Intelligence and Linguistics, aiming to enable computers to comprehend human language statements or words. Its inception was motivated by the desire to simplify user interactions with computers through natural language communication. NLP can be broadly categorized into two components: Natural Language Understanding or Linguistics, which focuses on comprehending language, and Natural Language Generation, which involves tasks related to creating text.

Natural language processing (NLP) was originally different from text information retrieval (IR), which uses efficient, statistics-based techniques to index and search large amounts of text. Over time, NLP and Information Retrieval (IR) have become somewhat similar. Today, NLP borrows from many different fields, so NLP researchers and developers need to have a wide range of knowledge (Nadkarni et al., 2011).

Given the significant proliferation of user-generated textual content on the Internet, researchers across various domains, particularly in the realm of Natural Language Processing, have shown considerable interest in the automated extraction of valuable information from the abundant corpus of documents (Sun et al., 2017, #).

Today, Natural language processing is being applied in many fields. Some of the popularly used NLP applications include; Machine translation i.e, translating technical manuals, support documents or catalogs. It is also being used for Automatic summarization where meanings or records and documents are summarized when a user tries to access relevant information from a large knowledge base. It is mostly used to provide a high level summary of blog posts etc (Malathi & Ambeth Kumar, 2021).

The primary goals of Natural Language Processing (NLP) encompass interpreting, analyzing, and manipulating natural language data to achieve specific purposes, employing a range of algorithms, tools, and methods. Nevertheless, numerous challenges arise, often contingent on the specific characteristics of the natural language data in question, making it challenging to fulfill all objectives through a singular approach. As a result, there has been significant research interest in the recent past dedicated to developing diverse tools and methods within the field of NLP and

related areas of study (Khurana et al., 2023, 3713-3744). The figure below shows the recent developments in NLP.

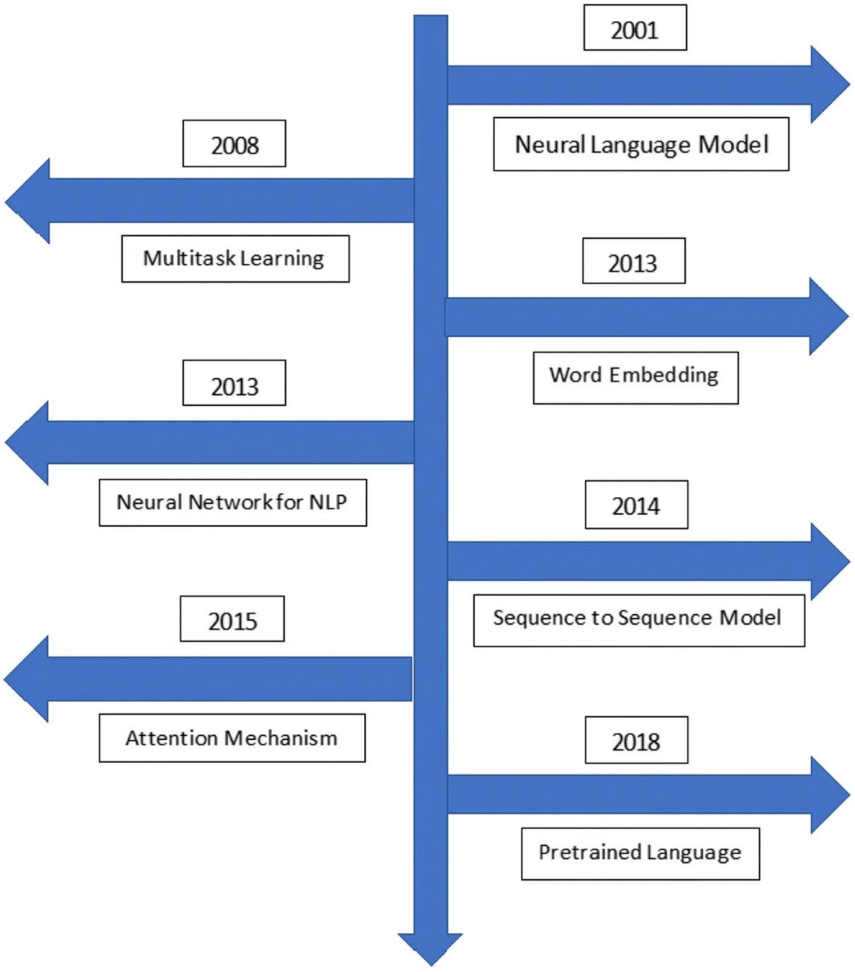


Figure 2.1: Recent developments in NLP

Natural language processing has also gained a lot of traction and is being used in the following industries; Sentiment analysis especially in social media platforms, Healthcare, Text mining, Education, and Agriculture (Malathi & Ambeth Kumar, 2021).

The problem of too much information in all areas, including business, healthcare, and education, has caused an increase in unstructured data, which is not considered useful. In this context, natural language processing (NLP) has so far proven to be an effective technology that can be used with

advanced technologies such as machine learning, artificial intelligence, and deep learning to improve the ability to understand and process natural language (Bahja, 2021).

2.3 Music Recommendation Systems

There is no doubt that the digital age changed the structure of the music industry. The internet and related technologies led to the creation of digital music, music streaming services and online radios which have all gained popularity over the years. This has been largely due to the convenience of accessing music through these platforms (Daniella & Capodulipo, 2015).

Advances in social networks and web 2.0 technology have led to the addition of features such as sharing playlists with friends, seeing what friends are listening to in addition to liking and rating the artists (Yue & Xi, 2011).

Music recommendation systems have become increasingly popular in recent years due to the vast amount of music available online (Celma & Schaefer, 2014). Natural language processing (NLP) techniques have been used to extract information from music-related text data to improve music recommendation systems. (Huang & Chen, 2015). One of the major challenges in music recommendation systems is dealing with the 'cold-start' problem, where little or no information is available about a new user or item. (Schedl & Knees, 2016).

There is a potential benefit of using lyrics as a source of information for music recommendations as lyrics contain rich information that can be used to improve the recommendations. Lyrics-based music recommendation systems take advantage of the rich information contained in song lyrics and have been shown to improve the effectiveness of music recommendations (Liu & Yang, 2018). Music recommendation systems are a promising application of natural language processing (NLP) due to the ability of NLP to extract information from text data such as lyrics, artist name and album name (Wang & Lin, 2019).

Some approaches to music recommendation have been discussed below:

a) **Collaborative filtering** is a method that uses the preferences of other users to make recommendations (Vihar, 2022). A collaborative filtering-based recommender system relies on evaluations of items submitted by a community of users (Gediminus & Tuzhilin,

2005, 734-749). It recommends items that the user in question has not yet considered but is likely to enjoy (Ricci et al., 2015, 1-34).

In fact, Collaborative Filtering employs the nearest neighborhood algorithm to identify the most similar items for a given user. Users with akin preferences for products, and who have rated similar items or products, are grouped in the same neighborhood. This proves beneficial in suggesting items that one user has not rated but have been rated by another user within the same neighborhood (Sharma et al., 2016). For example, if two users have similar preferences for songs with certain lyrics, and one of them likes a particular song, the system might recommend that song to the other user as well.

b) Content-based filtering

Content-based filtering is a method of filtering that relies on profile attributes. Recommendations are generated by examining user profiles, which include details and preferences. Preferences are shaped by the user's ratings, views, or purchases. A system employing this approach examines the user's profile for highly rated items and compares them with unrated items. Recommendations are then made based on this comparison, suggesting similar positively rated items to the user (Nair et al., 2021, 1-6).

In the context of music, the content-based filtering method uses the lyrics of the songs themselves to make recommendations (Shuvayan, 2015). For example, if a user likes songs with lyrics about a certain topic, the system might recommend other songs with similar lyrics.

c) Hybrid methods

This combines collaborative filtering and content-based filtering to make recommendations. (Verma & Yugesh, 2021). For example, the system might use both the preferences of other users and the lyrics of the songs to recommend songs to a user.

A hybrid approach within collaborative filtering and content-based filtering involves combining two or more techniques to enhance outcomes (Sharma et al., 2016). There are various criteria for applying a hybrid approach, including:

- Generating predictions by employing diverse content-based methods and subsequently consolidating the predictions (Sharma et al., 2016).
- Integrating the attributes of a content-based approach with a collaborative approach (Sharma et al., 2016).
- Combining the characteristics of a collaborative approach with a content-based approach (Sharma et al., 2016).
- Developing a comprehensive model that merges the attributes of a content-based method and a collaborative filtering method. This integration aims to mitigate filtering issues by leveraging the advantages of one technique to overcome the disadvantages of another (Sharma et al., 2016).

Chapter 3: Methodology

3.1 Introduction

For the purposes of this research, the (The Cross Industry Standard Process for Data Mining) CRISP-DM methodology was used. It will involve the steps in the diagram below:

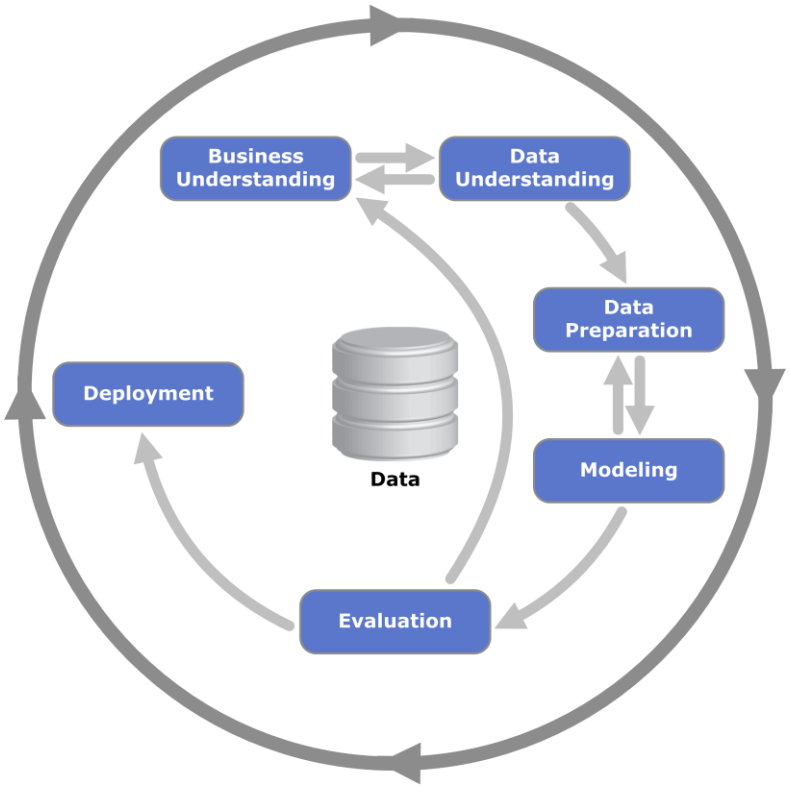


Figure 3.1: Crisp-DM methodology

This methodology attempts to reduce inherent project risk by allowing cycles where the project teams can go back to reassess requirements and firm up on the project understanding. This provides more ease-of-change during the development process.

This methodology was chosen because it is generalizable. William Vorhies, one of the creators of CRISP-DM, argues that because all data science projects start with business understanding, have data that must be gathered and cleaned, and apply data science algorithms, “CRISP-DM provides strong guidance for even the most advanced of today’s data science activities (Vorheis, 2016).

3.2 Data Source and Dataset

The dataset utilized in this study was sourced from Mendeley Data, which is a reputable and freely accessible cloud-based communal data repository. This platform serves as a secure repository for diverse data sets across various domains, facilitating easy access and sharing of research data among the scientific community.

The tcc_ceds_music dataset provides a list of songs from 1950 to 2019. It comprises a comprehensive collection of music-related data curated for research purposes in the domain of computational music analysis and recommendation systems. This dataset includes information such as song titles, genres, release dates, lyrics and other potentially relevant metadata such as sadness, danceability, loudness, acousticness, etc.

The most notable feature of the dataset for this study is its inclusion of song lyrics, which adds a textual dimension to the dataset. This aspect enables researchers to delve into the semantic content of songs, opening avenues for sentiment analysis, natural language processing and text-based recommendation systems.

Overall, the dataset serves as a valuable resource for researchers seeking to explore and analyze music-related phenomena, develop machine learning models for music recommendation, sentiment analysis or investigate the intersection of music and language.

According to (*Music Topics and Metadata*, 2020), the audio data was scraped using Echo Nest® API integrated engine with Spotify's python package. The spotify API permits the user to search for specific genres, artist songs, release dates etc. To obtain the lyrics, the Genius® API was used as the base URL for requesting data based on the song title and artist name.

	Unnamed: 0	artist_name	track_name	release_date	genre	lyrics	len	dating	violence	world/life	...	sadness	feelings	danceability	loudness	acousticness	instrumentalness	valence	energy
0	0	mukesh	mohabbat bhi jhoothi	1950	pop	hold time feel break feel untrue convince speak voice tear try hold hurt try forgive okay play break string feel heart want feel tell real truth hurt lie worse anymore little turn dust play house ruin run leave save like chase train late late tear try hold hurt try forgive okay play break string feel	95	0.000598	0.063746	0.000598	...	0.380299	0.117175	0.357739	0.454119	0.997992	0.901822	0.339448	0.137110

Figure 3.2: Sample row of the dataset and respective features

Below is a description of the variables that this research utilized in building the music recommender system.

Table 3.1: Data attributes and description

Attribute	**Description**	**Data type**
artist_name	The name of the artist or band who performed the song.	String
track_name	The title of the song.	String
release_date	The year the song was released	Int
genre	The genre or genres that best describe the song.	String
lyrics	The lyrics of the song	String
len	Contains the count of words in the lyrics column	Int

The data also contains some audio and metadata in the form of numerical features extracted from the audio signal of the songs i.e. danceability, loudness, acousticness, instrumentalness, valence, energy etc.

3.3 Data Pre-Processing

The next step in this research was to preprocess the data. This is an essential step in preparing raw data for analysis and modeling. It typically involves tasks such as handling missing values, removing duplicates, encoding categorical data etc.

Since we are working mostly with text data, the following steps were taken in the clean-up and preprocessing:

- To clean the lyrics columns, regular expressions were used to remove any text enclosed within square or round brackets, remove double newline characters, and replace single newline characters with a space.
- A language detection function was applied to determine the language of the lyrics. This was a crucial step in ensuring that we only work with songs whose lyrics are in English

```
[ ]  def lang_detector(x):
         '''
             takes a string and returns language of string
         '''
         try:
             return detect(x)
         except:
             return 'unknown language'
```

```
def cleaner(df):

    #cleans lyrics column using regex
    df['lyrics'] = df['lyrics'].apply(lambda x: re.sub("[\(\[].*?[\)\]]", "", x))
    df['lyrics'] = df['lyrics'].apply(lambda x: re.sub('\n\n', '', x))
    df['lyrics'] = df['lyrics'].apply(lambda x: re.sub('\n', ' ', x))

    #applying the language detector fucntion
    df['lang'] = df['lyrics'].apply(lambda x: lang_detector(x))
    df = df[df['lang'] == 'en']

    # #create new column for song length
    # df['song_length'] = df['lyrics'].apply(lambda x: len(x))


    Q1 = df['len'].quantile(0.25)
    Q3 = df['len'].quantile(0.75)
    IQR = Q3 - Q1
    df = df[(df['len'] > (Q1 - 1.5 * IQR)) & (df['len'] < (Q3 + 1.5 * IQR))]
    df['song_length_quantiles'] = pd.cut(df['len'], bins=10, precision=0)
    return df

df = cleaner(df)
```

Figure 3.3: Data Preprocessing

3.3.1 Text Preprocessing

This step involves the process of cleaning and transforming raw text data into a format that is suitable for natural language processing (NLP) tasks. For this particular study, the following steps were taken:

- Lowercasing - This step ensures that all the text in the lyrics column are converted into lowercase to ensure consistency
- Removing stop words - Stop words are common words that occur frequently in text but are insignificant to the overall meaning of the text. Removing stop words, helps to reduce dimensionality of the text data and focuses on more meaningful words.
- Removing noise - Noise in text data refers to unnecessary elements that can affect the performance of NLP tasks. This may include special characters, punctuation marks, HTML tags, or any other non-alphanumeric characters that do not carry significant meaning.
- Stemming or Lemmatization - This is a technique used to reduce words into their base or root forms.

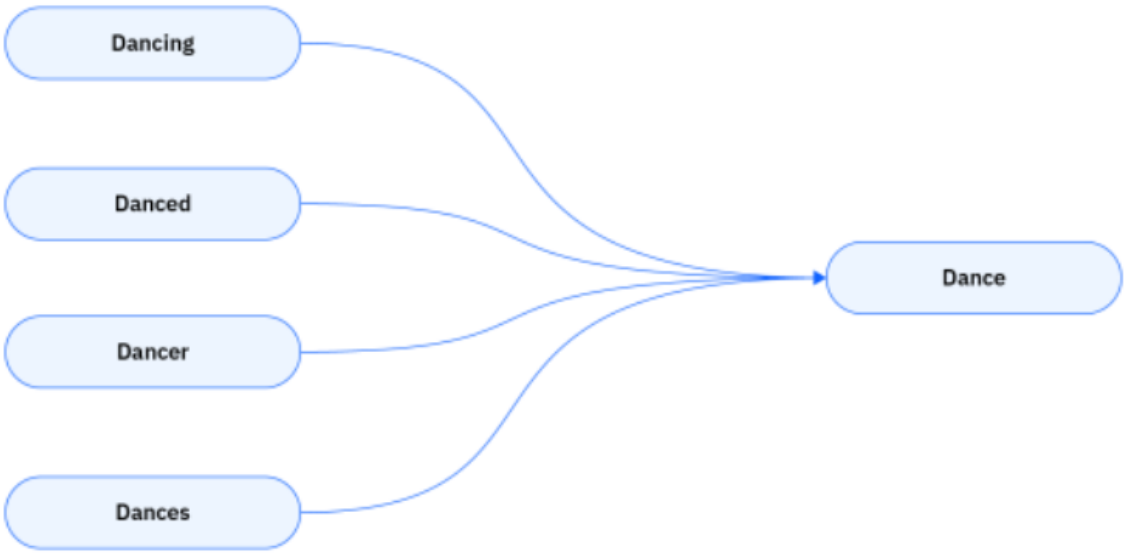


Figure 3.4: Root Word

In academic discourse, stemming is commonly understood as the act of removing affixes from words to generate stemmed word sequences, while lemmatization encompasses a broader process of reducing various morphological variations to a single dictionary base form. The key practical difference lies in their respective approaches: while stemming focuses on the removal of common suffixes from word tokens, lemmatization aims to produce output words that represent existing normalized forms, or lemmas, found in a dictionary (IBM, 2023).

For this study, the lemmatization approach was applied. It is useful for this study because we require high accuracy and precision in representing words in their base form by maintaining the semantic meaning of words and applying some text classification and sentiment analysis.

3.3.2 Feature Engineering

In natural language processing, feature engineering plays a pivotal role in transforming raw text data into structured numerical features that can be utilized by machine learning models. This phase encompasses a series of preprocessing and analysis tasks aimed at extracting meaningful information from textual data.

For this study, the following additional features were created from the data, which were:

- **Tokenization of lyrics** - This involves breaking down raw text data into their smallest meaningful units, known as tokens. A token is the smallest unit of a word. This is a fundamental process and is essential in Part-Of-Speech (POS) tagging within natural language processing (NLP)

- **Word count and word frequencies -** For this stage, the number of words present in each song lyrics column were quantified. Additionally, word frequencies were extracted inorder to provide insights into the relative prevalence of different words across the entire data set. These metrics were useful to identify common themes, topics or patterns within the dataset (Rai & Borah, 2021).

```python
#Here we want to show the most common words in our lyrics collection

from collections import Counter

# Function to tokenize lyrics into words
def tokenize_lyrics(lyrics):
    return lyrics.split()

# Tokenize lyrics into words
tokenized_lyrics = df['lyrics'].apply(tokenize_lyrics)

# Flatten the list of tokenized lyrics
all_words = [word for sublist in tokenized_lyrics for word in sublist]

# Count the frequency of each word
word_freq = Counter(all_words)

# Get the most common words and their frequencies
most_common_words = word_freq.most_common(10)  # Change 10 to the desired number of most common words

print("Most common words in the lyrics:")
for word, freq in most_common_words:
    print(f"{word}: {freq} times")
```

```
Most common words in the lyrics:
know: 29668 times
time: 24198 times
like: 22851 times
come: 20953 times
heart: 16161 times
away: 15547 times
go: 15267 times
feel: 14626 times
life: 14561 times
yeah: 13953 times
```

Figure 3.5: Word Frequency

- **Sentiment scores -** Sentiment analysis involves quantifying the emotional or subjective tone conveyed by the lyrics of each song. This process typically assigns sentiment scores to individual songs based on the polarity and intensity of emotions expressed. Sentiment scores can range from negative to positive, or they may be more nuanced, capturing a spectrum of emotions such as joy, sadness, anger, or neutrality. These scores offer valuable insights into the mood or sentiment conveyed by the lyrics, which can inform downstream applications such as music recommendation.

```
[ ]  # some sentiment analysis on the lyrics

     from textblob import TextBlob

     # Function to calculate sentiment polarity of lyrics
     def calculate_sentiment(lyrics):
         blob = TextBlob(lyrics)
         sentiment_score = blob.sentiment.polarity
         return sentiment_score

     # Calculate sentiment polarity for each song
     df['sentiment'] = df['lyrics'].apply(calculate_sentiment)

     # Display the sentiment polarity for each song
     print("Sentiment polarity for each song:")
     print(df[['track_name', 'sentiment']])
```

```
Sentiment polarity for each song:
                           track_name  sentiment
0               mohabbat bhi jhoothi  -0.096875
1                          i believe   0.450000
2                                cry  -0.110000
3                           patricia   0.150000
4                 apopse eida oneiro   0.134091
...                              ...        ...
28367               10 million ways  -0.042500
28368  ante up (robbin hoodz theory)   0.410000
28369                 whutcha want?   0.130952
28370                        switch   0.054167
28371                        r.i.p.  -0.161538

[26311 rows x 2 columns]
```

Figure 3.6: Sentiment analysis

3.4 Text Vectorization

Term Frequency - Inverse Document Frequency (TF-IDF) is a prevalent statistical technique employed text vectorization tasks in natural language processing. It assesses the significance of a term in a document compared to a set of documents known as a corpus.Text documents undergo a process of vectorization to assign numerical importance scores to words (Martin, 2023).

For this study, a numerical score was calculated for each word based on how often it appears in the lyrics i.e term frequency, and how rare it is across all the songs in the dataset i.e. inverse document frequency. This allows us to quantify the relevance of each word to a specific song.

The Term Frequency (TF) of a term or word refers to how many times the term occurs in a document relative to the total number of words in that document.

$$TF = \frac{number\ of\ times\ the\ term\ appears\ in\ the\ document}{total\ number\ of\ terms\ in\ the\ document}$$

The Inverse Document Frequency (IDF) of a term indicates the proportion of documents in the corpus that include the term. Terms that appear in only a small fraction of documents, such as specialized technical jargon, are assigned higher importance values compared to terms that are present in a large majority of documents.

$$IDF = log\ (\frac{number\ of\ documents\ in\ the\ corpus}{number\ of\ documents\ in\ the\ corpus\ that\ contain\ the\ term})$$

The TF-IDF of a term is calculated by multiplying the TF and IDF scores.

$$TF - IDF = TF * IDF$$

To implement the above, the python machine learning library SKlearn which has a TfidfVectorizer() function was used.

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

# Initialize TF-IDF vectorizer
tfidf_vectorizer = TfidfVectorizer()

# Fit and transform lyrics to TF-IDF vectors
tfidf_matrix = tfidf_vectorizer.fit_transform(df_subset['lyrics'])
```

Figure 3.7: TF-IDF Vectorizer

3.5 Building the Recommender System

3.5.1 Cosine Similarity

Cosine similarity gauges the similarity between two vectors within an inner product space. It quantifies this similarity by computing the cosine of the angle between the vectors, indicating whether they are aligned in a similar direction. This method is commonly employed to assess document similarity within text analysis (Jiawei et al., 2012).

Mathematically, cosine similarity is defined as the dot product of vectors divided by their magnitude (Martin, 2022). For example, if we have two vectors, A and B, the similarity between them is calculated as:

$$similarity\ (A, B)\ = cos\ (\theta)\ = \frac{A \cdot B}{||A||\ ||\ B\ ||}$$

Where:

- θ is the angle between the vectors,
- $A\ \cdot\ B$ is dot product between A and B and calculated as

  $$A\ \cdot\ B = A^T\ \ B\ =\ \sum\ \ \ \ A_i\ B_i =\ A_{\ 1}B_1\ +\ A_2\ B_2 + \ldots + A_n\ B_n$$

- $||A||$ represents the L2 normalization or magnitude of the vector which is calculated as $||A||$ = $||A||\ =\ \sqrt{A_1^2\ + A_1^2 + A_1^n}$

For this study, cosine similarity was applied using the SKlearn cosine similarity python library.

The process of combining TF-IDF and cosine similarity implemented by following the steps below:

- Calculate the term frequency (TF) for each word
- Calculate the inverse document frequency (IDF) for each word

- Multiply the TF and IDF values for each word to get the TF-IDF weight.
- Create a vector for each piece of text by using the TF-IDF weights as the values for the vector.
- Calculate the cosine similarity between the two vectors.

The resulting cosine similarity value is a number between -1 and 1, where 1 represents identical vectors and -1 represents vectors pointing in opposite directions. The higher the cosine similarity value, the more similar the two pieces of text are considered to be.

This approach leverages the idea that if two songs have similar lyrics, they are likely to appeal to similar audiences, even if they have not been explicitly rated together. By using TF-IDF and cosine similarity, we can recommend songs that share thematic or lyrical similarities, providing users with personalized music recommendations based on their preferences.

3.5.2 Levenshtein Distance

The Levenshtein distance is a metric measuring the dissimilarity between two strings and quantifies the minimum number of single-character operations (e.g., deletion, insertion, or substitution) needed to convert one string (s1) into another string (s2) (Dutta et al., 2022, 297-303). It measures the minimum number of single-character edits required to change one string into another. It's often used in natural language processing and spell checking to quantify the similarity between two strings.

To enhance the recommendation, Levenshtein distance matrix was added as a preprocessing step to identify potential synonyms or misspellings in song titles. This was to help improve the quality of recommendations by ensuring that similar songs with slightly different titles are considered.

The following steps were taken:

1. Calculating Levenshtein distance matrix for song titles

2. Identifying potential synonyms or misspellings to ensure that similar songs with slightly but different titles are considered in the recommendations.

3. Finally, we used the information from the Levenshtein distance matrix to enhance recommendations obtained from the TF-IDF and cosine-similarity approach.

For this study, Levenshtein distance did not identify many potential synonyms or misspellings in the song titles, indicating that the titles were already relatively consistent and similar enough to generate similar recommendations.

3.5.3 Ranking Music

The model works by retrieving the row of cosine similarity scores corresponding to the target song from the list. It then sorts the similarity scores in descending order and excludes the target song. Finally, it extracts the indices of the most similar songs up to the specified number of recommendations; in this case, 5.

3.6 Performance Evaluation

In this section of the study, we present the evaluation of the music recommendation system. The primary objective of this phase is to assess the performance of the TF-IDF and Cosine Similarity combination in recommending songs based on their similarity to a target song.

The outcome of the model evaluation phase is crucial for determining the effectiveness of the music recommendation system. The following metrics were analyzed:

- Accuracy - this metric measures how often a machine learning model correctly predicts the outcomes and is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precision - this is defined as the ratio of correctly classified positive samples to the total number of classified positive samples.

$$Precision = \frac{TP}{TP + FP}$$

- Recall - this measures the proportion of positive samples that were actually identified correctly.

$$Recall = \frac{TP}{TP + FN}$$

- F1 score - this metric is calculated as the harmonic mean of precision and recall.

  - $F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall}$

    Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives

3.7 Deployment

For deployment, Streamlit was used. This is an open-source Python Library that allows one to create interactive web applications for machine learning, data science, and other computational tasks. Chapters 4 and 5 give more details on the system and its implementation.

Chapter 4: System Design and Architecture

4.1 Workflow

The music recommendation system is designed to provide personalized song recommendations based on the analysis of song lyrics. The system architecture follows a modular design, consisting of several key components: data collection, preprocessing, feature extraction, recommendation engine, and user interface.

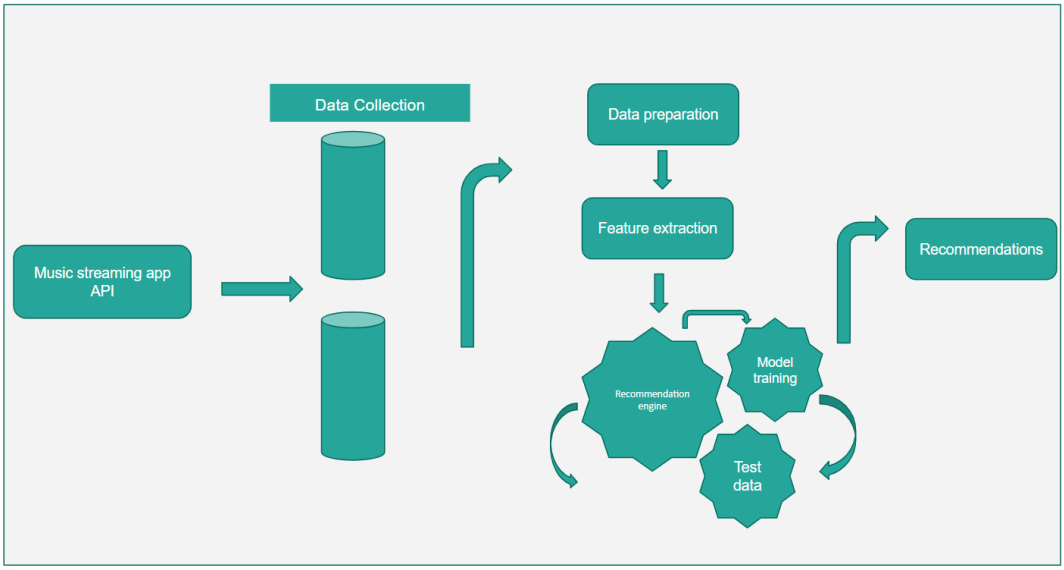


Figure 4.1: System architecture and design

The data collection component retrieves song lyrics and metadata from online music databases obtained through the use of APIs. This platform serves as a secure repository for diverse data sets across various domains, facilitating easy access and sharing of research data among the scientific community.

The preprocessing or data preparation module cleans and prepares the raw data for analysis by removing noise, handling missing values, and standardizing the format. Feature extraction involves transforming the song lyrics into numerical representations using techniques such as TF-IDF (Term Frequency-Inverse Document Frequency) which is discussed in more detail in the following sections of this paper. These features are then used by the recommendation engine to generate song recommendations for users.

The app works by retrieving a list of song names loaded from the song dataset. The user is then prompted to select a song from the list of available songs. The selected song is matched to its corresponding index in the dataset and the index is stored. The recommendations are then retrieved and matched to the original song based on their cosine similarity scores. The application then presents to the user the top five recommended songs based on the selected song, along with their details.

The use case diagram below summarizes the interaction of the users with the system.

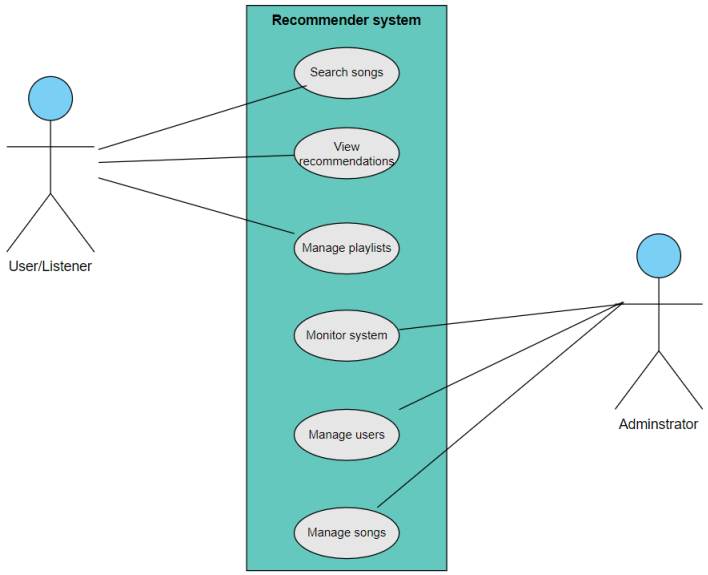


Figure 4.2: Use case diagram

Chapter 5: System Implementation and Testing

5.1 System Implementation

Below is a brief overview of the system design and architecture of the web app created using Streamlit:

5.1.1 Frontend

- Streamlit generates the frontend of the web app dynamically based on the Python code written. The layout and user interface are defined as elements using Python functions and decorators provided by Streamlit.
- The frontend consists of interactive widgets like sliders, buttons, checkboxes, and text inputs, which allow users to control the behavior of the app.
- Streamlit automatically updates the frontend in real-time as users interact with the app, without needing to write any additional code for handling user input or state management.

5.1.2 Backend

- Streamlit manages the backend of the web app behind the scenes, handling communication between the frontend and the Python code running on the server.
- When a user interacts with the app (e.g., selects a value or clicks a button), Streamlit triggers the corresponding Python function or callback defined in the app code.
- The Python code processes the user input, executes any necessary computations or data manipulations, and generates the updated content to be displayed on the frontend.
- Streamlit then sends the updated content back to the frontend, where it is rendered and displayed to the user.

The figures below show the front-end and back-end implementation of the web application.

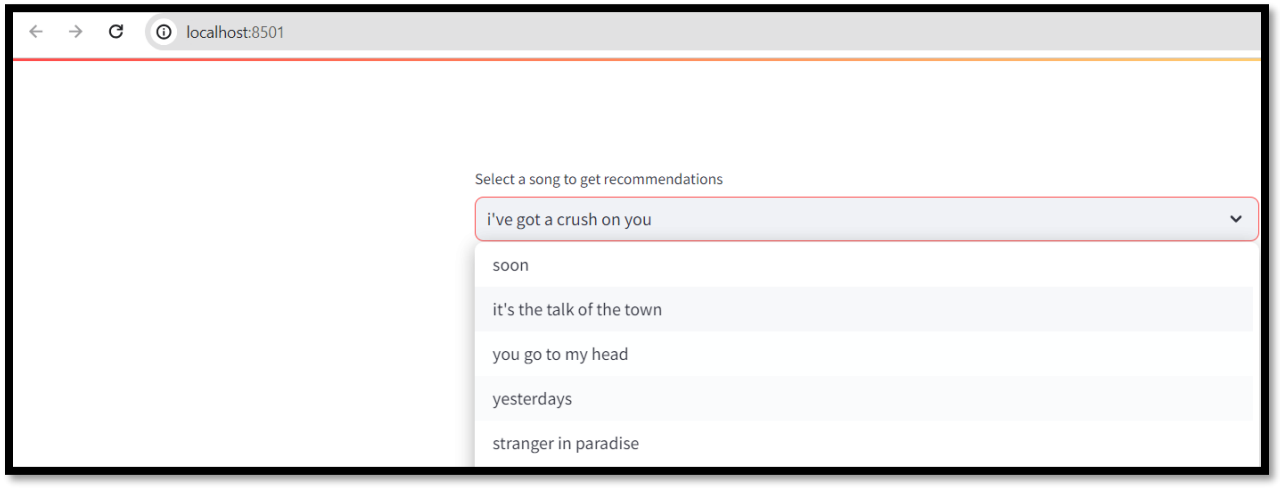
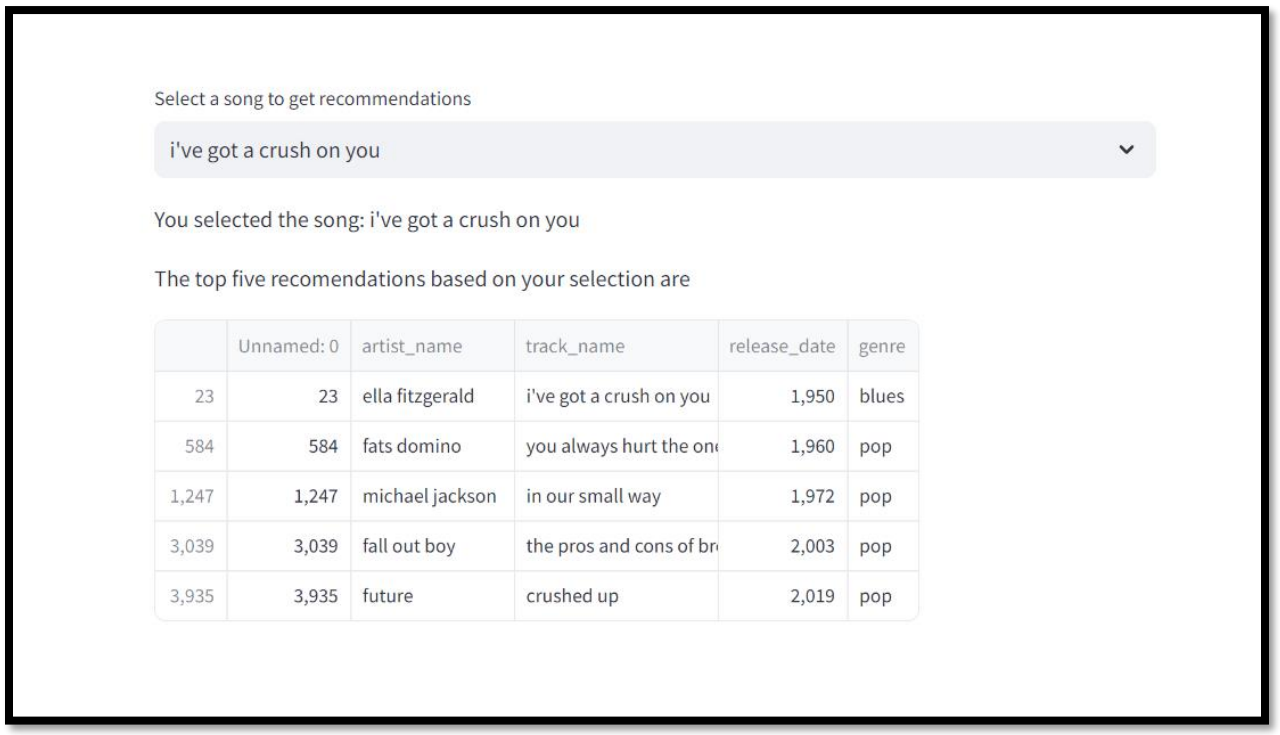


Figure 5.1: Song selection



	Unnamed: 0	artist_name	track_name	release_date	genre
23	23	ella fitzgerald	i've got a crush on you	1,950	blues
584	584	fats domino	you always hurt the one	1,960	pop
1,247	1,247	michael jackson	in our small way	1,972	pop
3,039	3,039	fall out boy	the pros and cons of br	2,003	pop
3,935	3,935	future	crushed up	2,019	pop

Figure 5.2: Streamlit recommendation output sample

server.py

C: > Users > njeri > Documents > School > Dissertation NLP > Streamlit app > server.py > ...

```python
import streamlit as st
import pandas as pd

import os

# Get the current working directory
current_dir = os.getcwd()
# Relative path to the CSV file
songs_path = os.path.join(current_dir, 'data','songs.csv')
recommendations_path = os.path.join(current_dir, 'data','recommendations.csv')
#import the datasets
songs = pd.read_csv(songs_path)
recommendations = pd.read_csv(recommendations_path)

#get a list of the track names
song_list = songs['track_name']

#Ask the user to select a song
selected_song = st.selectbox("Select a song to get recommendations",song_list)

#get the index the user selected
selected_index = songs[songs['track_name']==selected_song].index[0]

#get the songs matching that index
recommended_indexes = recommendations[recommendations['index']==selected_index]['Subject'].to_list()

#get songs in the recommeded indexes
recommended_songs = songs[songs.index.isin(recommended_indexes)]

#show the user the song they selected
st.write(f'You selected the song: {selected_song}')


#Output song recommendations

st.write('The top five recomendations based on your selection are ')

st.write(recommended_songs)
```

Figure 5.3: Back-end implementation of Streamlit application

Chapter 6: Discussion of Results

6.1 Exploratory Data Analysis

For this study, we engaged in an iterative exploration of the dataset to reveal underlying patterns, trends, and characteristics.

6.1.1 Univariate Data Analysis

This involves analyzing individual variables in isolation to understand their distribution, central tendency, and variability.

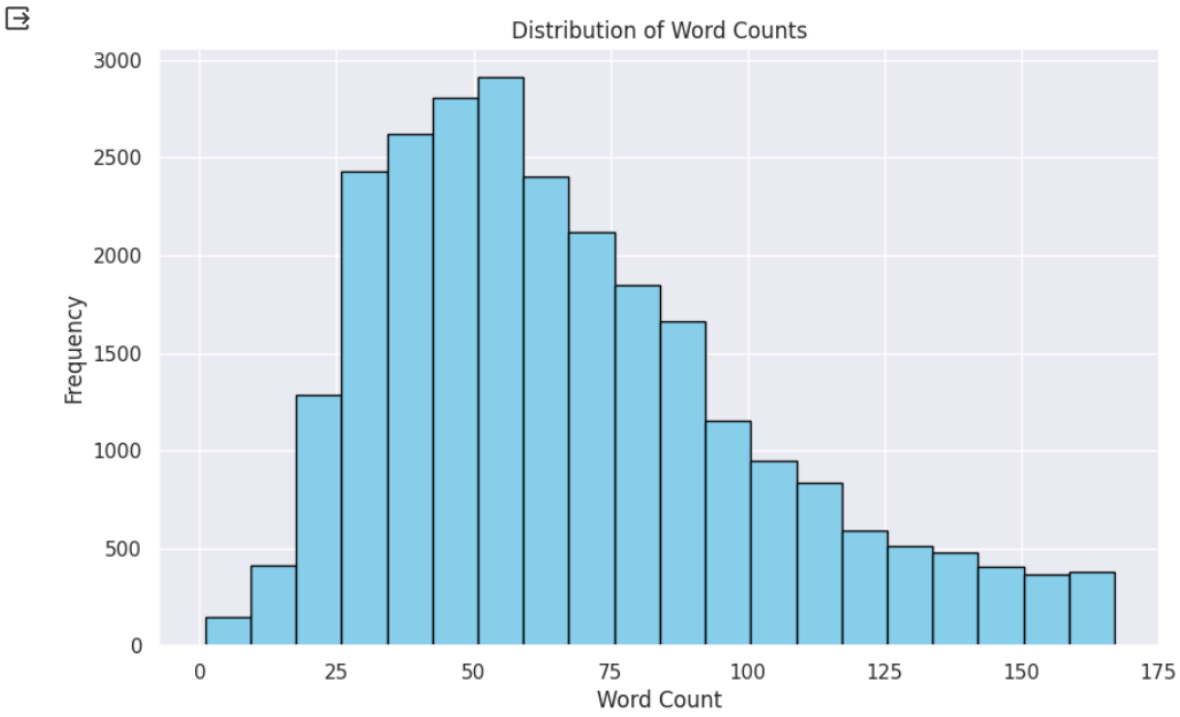


Figure 6.1: Word Counts

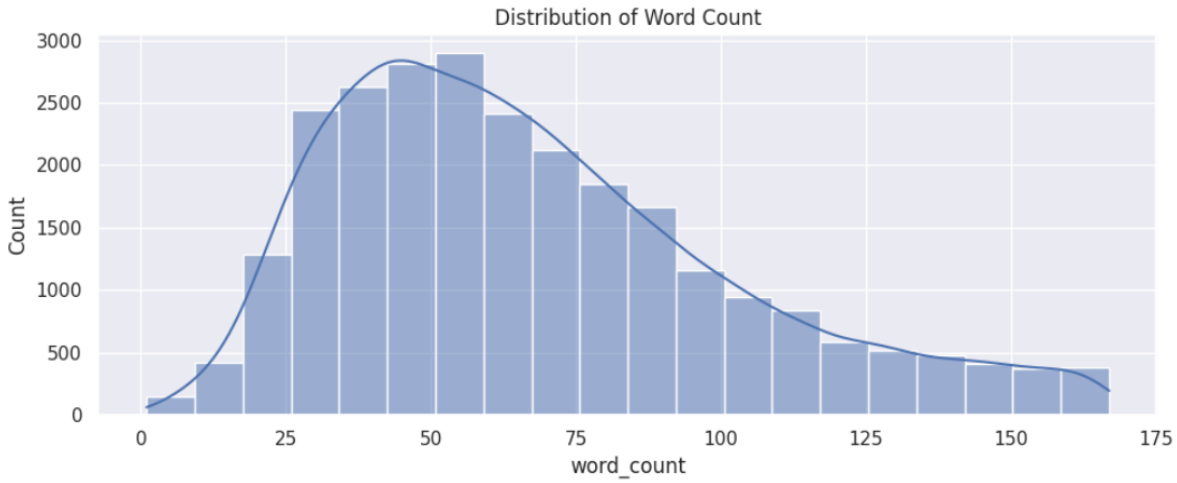


Figure 6.2: Word Counts distribution

The x-axis represents the range of word counts per song. The Y-axis displays the frequency count of the songs falling within each word count range. The histogram of word counts is right-skewed with a peak towards lower word counts. This indicates that the majority of the songs have relatively fewer words.

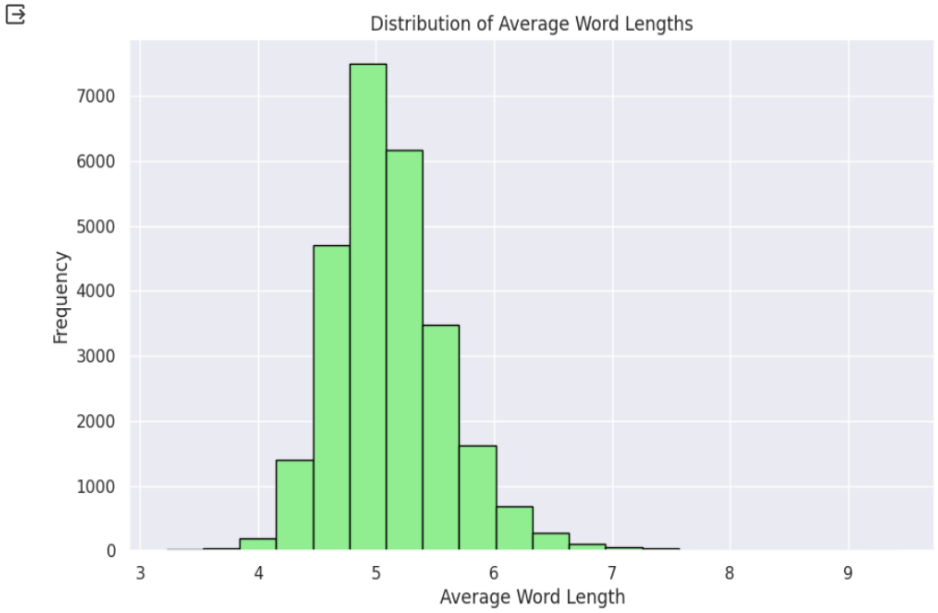


Figure 6.3: Word Length

Figure 6.4: Sentiment Scores Distribution

The X-axis represents the sentiment scores which range from negative 1 to positive 1. The Y-axis displays the frequency count of lyrics falling within each sentiment score range. The histogram of our data displays a generally symmetric distribution around zero, indicating that the data has a balanced mix of positive and negative sentiments.

6.1.2 Bivariate Data Analysis

This section of the study aims to explore the relationship between two variables. I.e. sentiment scores and genres.

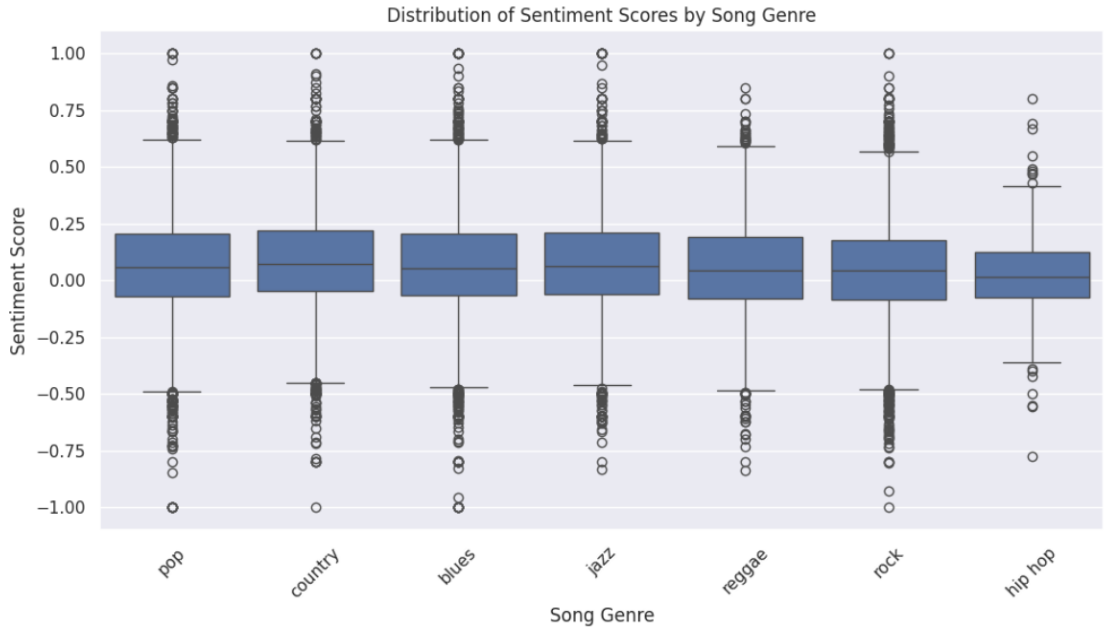


Figure 6.5: Bivariate data analysis

All the genres displayed a wide range of sentiment scores, indicating a diverse range of emotional expressions. The different genres also tended to have similar sentiment profiles. This suggested there could be some potential connections or overlaps between their emotional content.

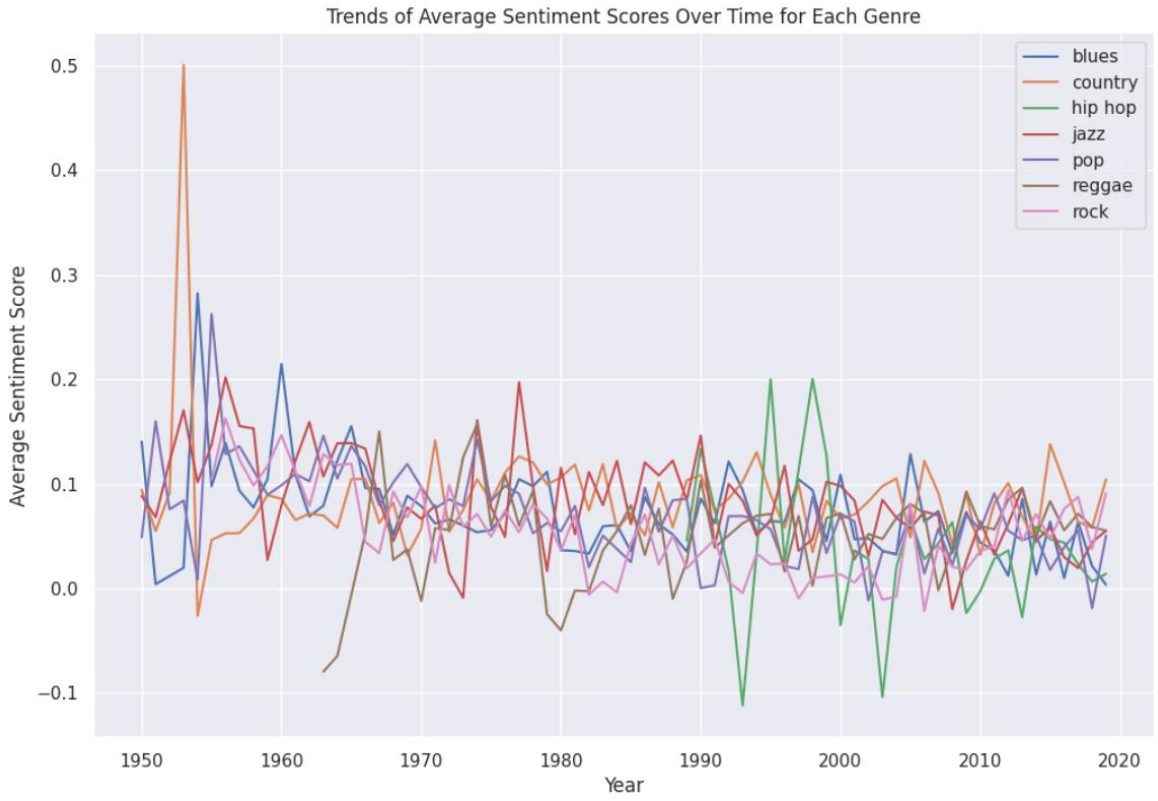


Figure 6.6: Sentiment trends over time for each genres

The above analysis of sentiment trends provides valuable insights into the emotional content of song lyrics within different genres and offers a glimpse into the evolving landscape of popular music over time.

Notably, the sentiment scores exhibit varying patterns across different genres. For instance, the sentiment scores for the “Rock”, “Country” and “Hip-hop” genres display more fluctuating patterns, with peaks and troughs occurring at irregular intervals over the years.

Moreover, the “Blues” genre exhibits a distinct downwards trend in sentiment scores, particularly in the last decade, suggesting a reducing positivity in the lyrical content of blues songs.

6.1.3 Word Cloud

We further examined the data using the visual below, where we generated a word cloud of at least 500 most common words found in the lyrics dataset.

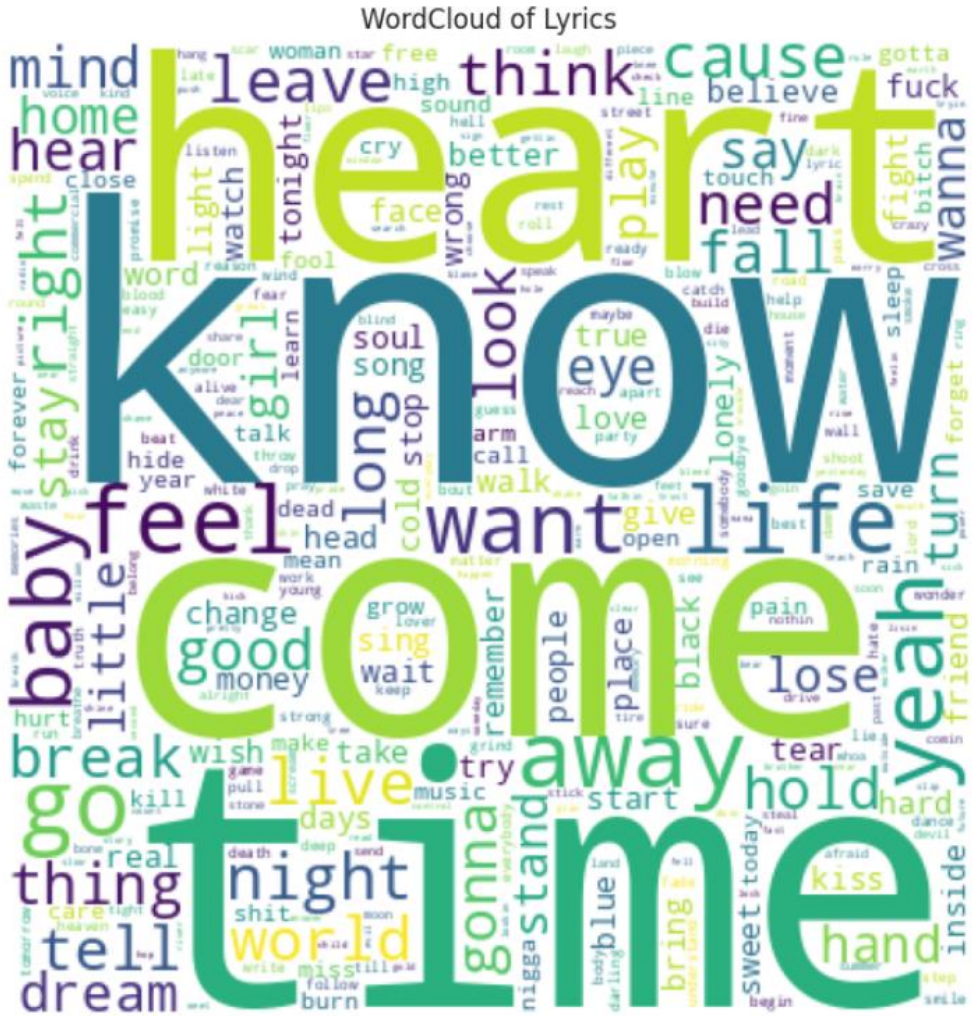


Figure 6.7: Word Cloud

The word cloud generated from the lyrics dataset offers a visual representation of the most prevalent themes present in the corpus of songs. The size of each word in the cloud is proportional to its frequency of occurrence in the lyrics.

By examining the word cloud, we identified prominent topics, emotions commonly expressed across the collection of songs. The word cloud also highlights the recurring keywords or phrases like “heart”, “know”, “come” and “time” which may hold significance within the context of the music genre or cultural trends.

6.2 Music Recommendation System Performance Results

For this study, each song was treated as an item and computed similarities between them based on their TF-IDF representations. Cosine similarity was used to measure the similarity between two songs' TF-IDF vectors. Thereafter, the cosine of the angle between the two vectors was computed, indicating how similar the songs are based on their word usage. Then, for a given target song, we can recommend similar songs by identifying those with the highest cosine similarity scores.

```
[ ]  # Create a DataFrame from the cosine similarity matrix
     cosine_sim_df = pd.DataFrame(cosine_sim, index=df_subset.index, columns=df_subset.index)

     # Print the shape of the cosine similarity matrix
     print("Shape of cosine similarity matrix:", cosine_sim_df.shape)

     # Print the cosine similarity matrix
     print("Cosine Similarity Matrix:")
     print(cosine_sim_df)
```

```
Shape of cosine similarity matrix: (3966, 3966)
Cosine Similarity Matrix:
          0         1         2         3         4         5         6     \
0     1.000000  0.004175  0.093175  0.034236  0.025271  0.071013  0.062016
1     0.004175  1.000000  0.000000  0.016794  0.005869  0.024326  0.124597
2     0.093175  0.000000  1.000000  0.027647  0.016770  0.035733  0.014552
3     0.034236  0.016794  0.027647  1.000000  0.014619  0.016632  0.073004
4     0.025271  0.005869  0.016770  0.014619  1.000000  0.004617  0.025644
```

Figure 6.8: Cosine Similarity Matrix

```
# Print recommended songs with their titles, artists, indices, genres, and cosine similarity scores
print("Recommended Songs:")
for i, (song_index, song_title, artist, genre) in enumerate(recommended_songs):
    print(f"{i+1}. Song Index: {song_index}, Title: {song_title}, Artist: {artist}, Genre: {genre}, Cosine Similarity: {similarity_scores[i]:.4f}")
```

```
Target Song:
Song Index: 3724
Title: goodnight chicago
Artist: rainbow kitten surprise
Genre: pop

Recommended Songs:
1. Song Index: 3105, Title: shut up!, Artist: simple plan, Genre: pop, Cosine Similarity: 0.4740
2. Song Index: 2908, Title: freakish, Artist: saves the day, Genre: pop, Cosine Similarity: 0.3756
3. Song Index: 718, Title: shut down (mono), Artist: the beach boys, Genre: pop, Cosine Similarity: 0.3124
4. Song Index: 3330, Title: shut up and let me go, Artist: the ting tings, Genre: pop, Cosine Similarity: 0.2980
5. Song Index: 3073, Title: the good times are killing me, Artist: modest mouse, Genre: pop, Cosine Similarity: 0.2268
```

Figure 6.9: Recommendation output sample

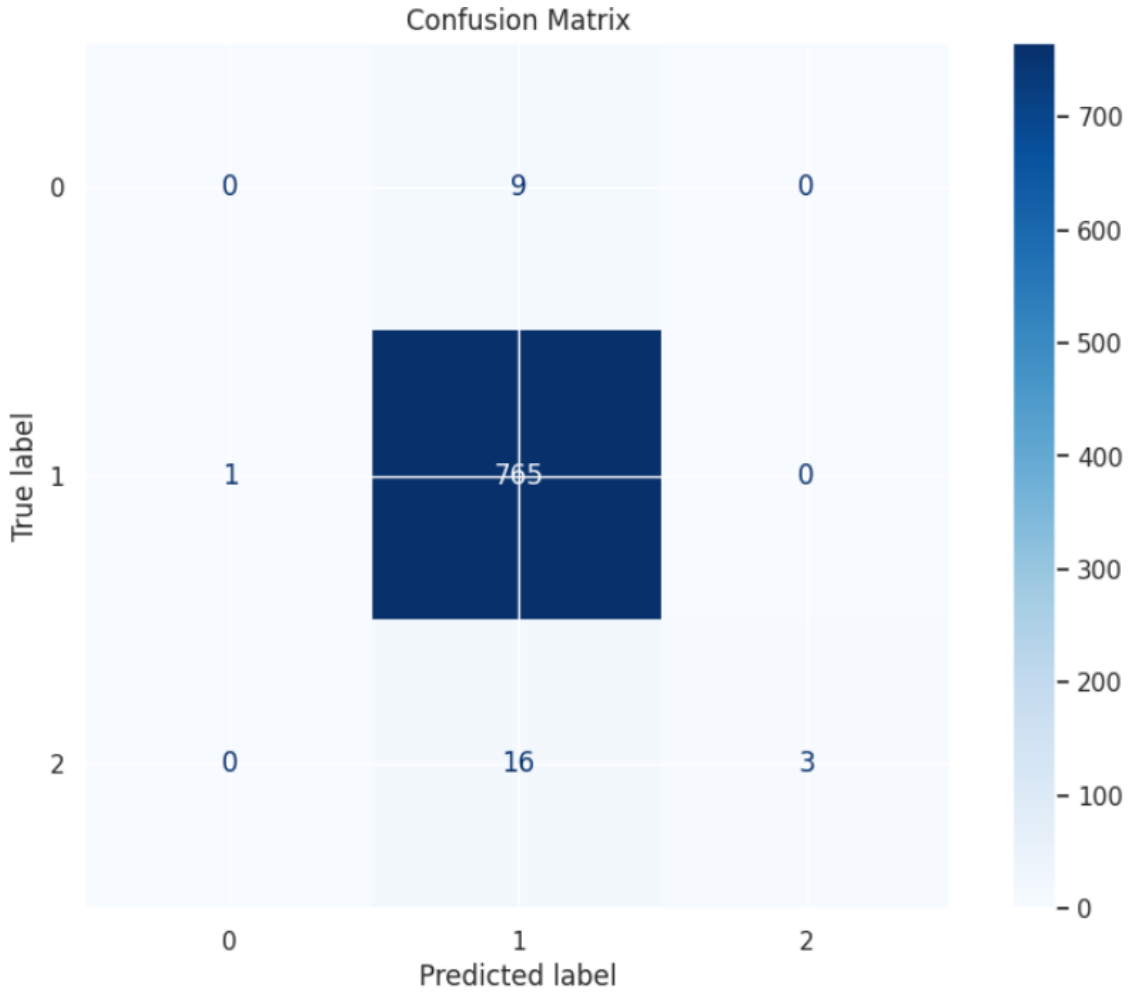


Figure 6.10: Confusion Matrix

The system performed generally well with the outputs below:

Model Evaluation Metrics:

Accuracy: 0.9673

Precision: 0.9581

Recall: 0.9673

F1 Score: 0.9551

The accuracy metric, which measures the overall correctness of the system's predictions, indicates that our system correctly identifies the recommended songs approximately 96.73% of the time.

This high accuracy rate suggests that the system is proficient at making accurate recommendations to users.

Precision, which measures the proportion of correctly recommended songs among all songs predicted as recommendations, is calculated to be 0.9581. This metric indicates that when the system recommends a song, there is a 95.81% probability that the song will be relevant and aligned with the user's preferences.

Similarly, the recall metric, which quantifies the proportion of relevant songs that are correctly identified by the system, is found to be 0.9673. This means that our system successfully captures approximately 96.73% of all relevant songs, ensuring that users are presented with a comprehensive selection of songs that match their tastes.

The F1 score provides a balanced assessment of the system’s performance. With an F1 score of 0.9551, it achieves a strong balance between precision and recall, indicating robust performance across both metrics.

Overall, the high values of these evaluation metrics demonstrate the effectiveness and reliability of our music recommendation system in accurately identifying and recommending songs that are likely to appeal to users.

Chapter 7: Conclusions, Recommendations and Future works

7.1 Conclusions of the Study

Overall, the activities carried out in this study demonstrate the feasibility and effectiveness of using NLP techniques such as TF-IDF and cosine similarity for music recommendation purposes. By leveraging textual data from song lyrics, we were able to develop a recommendation system that can provide users with relevant and personalized song suggestions. This approach has significant implications for the music industry, as it can enhance user engagement and satisfaction by delivering tailored music recommendations based on their preferences and tastes. Moreover, our study highlights the potential of NLP techniques to extract meaningful insights from unstructured text data and apply them to real-world applications such as music recommendation systems.

7.2 Future Works

One avenue for further research in this paper is to explore alternative machine learning algorithms. While this study focused on TF-IDF and cosine similarity, future research could investigate the performance of alternative techniques such as deep learning models or ensemble methods, in music recommendation systems. By examining their strengths and weaknesses, the insights gained can be used to identify optimal approaches for diverse music datasets.

Another avenue for further research would be to explore the development of multilingual recommendation systems capable of accommodating users with diverse language preferences. By analyzing and processing music lyrics in multiple languages, recommendation systems can cater to a global audience and provide personalized recommendations based on users' language proficiency and cultural backgrounds. For instance, partnering with research communities that are working towards advancing natural language processing (NLP) and machine learning (ML) for African languages.

References

Alawi, A. (2023, June 14). *The Psychology of Sound: How Music Impacts Emotions.* Retrieved January 21, 2024 from Mello Studio: https://www.mellostudio.com/en/psychology-of-sound/

Bahja, M. (2021). Natural Language Processing Applications in Business.

Celma, O., & Schaefer, R. (2014). Music recommendation and discovery in the long tail. *IEEE.Transactions on Emerging Topics in Computing, 2*(2), 170-179.

Chowdhruy, & Gobinda. (2003). Natural Language Processing. Natural Language Processing. Annual Review of Information Science and Technology.

Curry, D. (2024). *Home App Data Music Streaming App Revenue and Usage Statistics (2024).* Retrieved January 22, 2024 from Business of Apps: https://www.businessofapps.com/data/music-streaming-market/

Daniella, & Capodulipo. (2015). The Future of Music Marketing (Masters Thesis). *IEEE.Transactions on Emerging Topics in Computing*.

Dutta, S., Das, A. K., Ghosh, S., & Samanta, D. (2022). *Data Analytics for Social Microblogging Platforms.* Elsevier Science.

Gediminus, A., & Tuzhilin, A. (2005). Toward the next generation of recommender systems. *A survey of the state-of-the-art and possible extensions, 17*, 734-749. From https://ieeexplore.ieee.org/abstract/document/1423975?casa_token=eeW9YXi1_HoAAAAA:FMir9K6JHi6q3jb4YNPU7FJAWMlNf0Dkek-NCJVG9HpkCXEp6nYAUXLWtc3EF7Plp10Q5L0uuaU

Greenberg, D. M., & Rentfrow, P. J. (2017). Music and big data: a new frontier. *urrent Opinion in Behavioral Sciences, 18*, 50-56. From https://ieeexplore.ieee.org/abstract/document/9103378?casa_token=QPbirjUfX1wAAAAA:q2KUs2I3dc5dqQ3omP09Wn0tGjgqiIfQ_8ZlcNZXmMOObZ4CsUH6lVzzRPzq7hVWhDvZ-QVwCwc

How Collaborative Filtering Works in Recommender Systems. (n.d.). Retrieved April 1, 2024 from Turing: https://www.turing.com/kb/collaborative-filtering-in-recommender-system

Huang, S., & Chen, Y. (2015). Building a music recommendation system based on lyrics and genre. *Journal of Ambient Intelligence and Humanized Computing, 6*(11), 1-11.

Hujran, O., Durani, U., & Dmour, N. (2020). Building a music recommendation system based on lyrics and genre. Big Data and its Effect on the Music Industry.

IBM. (2023, December 10). *What Are Stemming and Lemmatization?* Retrieved March 27, 2024 from IBM: https://www.ibm.com/topics/stemming-lemmatization

Inellipaat. (2022). *10 Big Data Applications in Real Life - Updated 2022.* From https://intellipaat.com/blog/10-big-data-examples-application-of-big-data-in-real- life

Jafri, I. (2023). *What is Manhattan Distance in machine learning?* Retrieved April 4, 2024 from Educative.io: https://www.educative.io/answers/what-is-manhattan-distance-in-machine-learning

Jiawei, H., Kamber, M., & Pei, J. (2012). Data Mining Concepts and Techniques. *Information and Software Technology, 2019*.

Khurana, D., Koli, A., & Khatter, K. (2023). Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications, 82*(3), 3713-3744. From https://link.springer.com/article/10.1007/s11042-022-13428-4

Lawendowski, R., & Bieleninik, L. (2017). Identity and self-esteem in the context of music and music therapy: a review. *Health psychology report, 5(2)*, 85-99.

Lekamge, S., Marasignhe, A., Kalansooriya, P., & Nomura, S. (2017). A Visual Interface for Emotion based Music Navigation using Subjective and Objective Measures of Emotion Perception. *International Journal of Affective Engineering, 15*, 205-211. From https://www.jstage.jst.go.jp/article/ijae/15/2/15_IJAE-D-15-00039/_article/-char/ja/

Liu, Y., & Yang, H. (2018). music recommendation system based on deep recurrent neural network and song lyrics. *Journal of Ambient Intelligence and Humanized Computing, 9*(3), 1051-1067.

Malathi, S., & Ambeth Kumar, V. (2021). Smart Intelligent Computing and Communication Technology. Smart Intelligent Computing and Communication Technology.

Martin, B. (2022). *Cosine Similarity – LearnDataSci.* Retrieved April 1, 2024 from LearnDataSci: https://www.learndatasci.com/glossary/cosine-similarity/

Martin, B. (2023). *TF-IDF — Term Frequency-Inverse Document Frequency – LearnDataSci.* Retrieved April 1, 2024 from LearnDataSci: https://www.learndatasci.com/glossary/tf-idf-term-frequency-inverse-document-frequency/

Meteren, R. V., & Someren, M. V. (2000). Using content-based filtering for recommendation. *n Proceedings of the Machine Learning in the New Information Age*, 47-56. From https://users.ics.forth.gr/~potamias/mlnia/paper_6.pdf

Music Topics and Metadata. (2020, August 22). Retrieved March 25, 2024 from Mendeley Data: https://data.mendeley.com/datasets/3t9vbwxgr5/1

Nadkarni, Prakash, Machado, L., & Chapman, W. (2011). Natural Language Processing: An introduction. *American Medical Informatics Association, 18*(5), 544-551.

Nair, A., Paralkar, C., Pandya, J., Chopra, Y., & Krishnan, D. (2021). Comparative Review on Sentiment analysis-based Recommendation system. *6th International Conference for Convergence in Technology (I2CT)*, 1-6. From https://ieeexplore.ieee.org/abstract/document/9418222

Nikolsky, A., & Buraco, A. B. (2023). The evolution of human music in light of increased prosocial behavior: a new model. *Physics of Life Reviews*. From https://www.sciencedirect.com/science/article/abs/pii/S1571064523002142#:~:text=Our%20main%20claim%20is%20that,chains%20of%20teaching%20and%20learning.

Rai, A., & Borah, S. (2021). Study of Various Methods for Tokenization. *Applications of Internet of Things*, 193-200. From https://link.springer.com/chapter/10.1007/978-981-15-6198-6_18#citeas

Ricci, F., Rokach, L., & Shapira, B. (Eds.). (2015). *Recommender Systems Handbook.* Springer US. From https://link.springer.com/chapter/10.1007/978-1-4899-7637-6_1#citeas

Schäfer, T., Sedlmeier, P., Städtler, C., & Huron, D. (2013). The psychological functions of music listening. *Frontiers in Psychology, 4*. From https://www.frontiersin.org/articles/10.3389/fpsyg.2013.00511/full

Schedl, M., & Knees, P. (2016). "State-of-the-art in music recommendation: a meta-analysis. *User Modeling and User-Adapted Interaction, 26*(1), 59-91.

Sharma, S., Sharma, A., Sharma, Y., & Bhatia, M. (2016). Recommender System using Hybrid Approach. *International Conference on Computing Communication and Automation*.

Shuvayan, D. (2015). *Beginners Guide to learn about Content Based Recommender Engine.* From https://www.analyticsvidhya.com/blog/2015/08/beginners-guide-learn-content-based-recommender-systems/ Huang, S and Y Chen. 2015. "Building a music

recommendation system based on lyrics and genre." Journal of Ambient Intelligence and Humanized Computing 6(

Sun, S., Luo, C., & Chen, J. (2017). A review of natural language processing techniques for opinion mining systems,. *Information Fusion, 36,* 10-25. From https://www.sciencedirect.com/science/article/abs/pii/S1566253516301117

Turner, David, Schroeck, M., & Schockley, R. (2013). Analytics: The real-world use of big data in financial services. IBM Global Business Services.

Verma, & Yugesh. (2021). *A Guide to Building Hybrid Recommendation Systems for Beginners.* From https://analyticsindiamag.com/a-guide-to-building-hybrid-recommendation-systems- for-beginners

Vihar, K. (2022). *What Is Collaborative Filtering: A Simple Introduction.* From https://builtin.com/data-science/collaborative-filtering-recommender-system

Vorheis, W. (2016, July 26). *CRISP-DM – a Standard Methodology to Ensure a Good Outcome - DataScienceCentral.com.* Retrieved January 16, 2024 from Data Science Central: https://www.datasciencecentral.com/crisp-dm-a-standard-methodology-to-ensure-a-good-outcome/

Vuust, P., Heggli, O. A., Friston, K. J., & Kringelbach, M. L. (2022). Music in the brain. *Nature Reviews Neuroscience*(23), 287 - 305. From https://nature.com/articles/s41583-022-00578-5

Wang, W., & Lin, Y. (2019). A review of music recommendation methods based on natural language processing." ACM Comput. Surv.

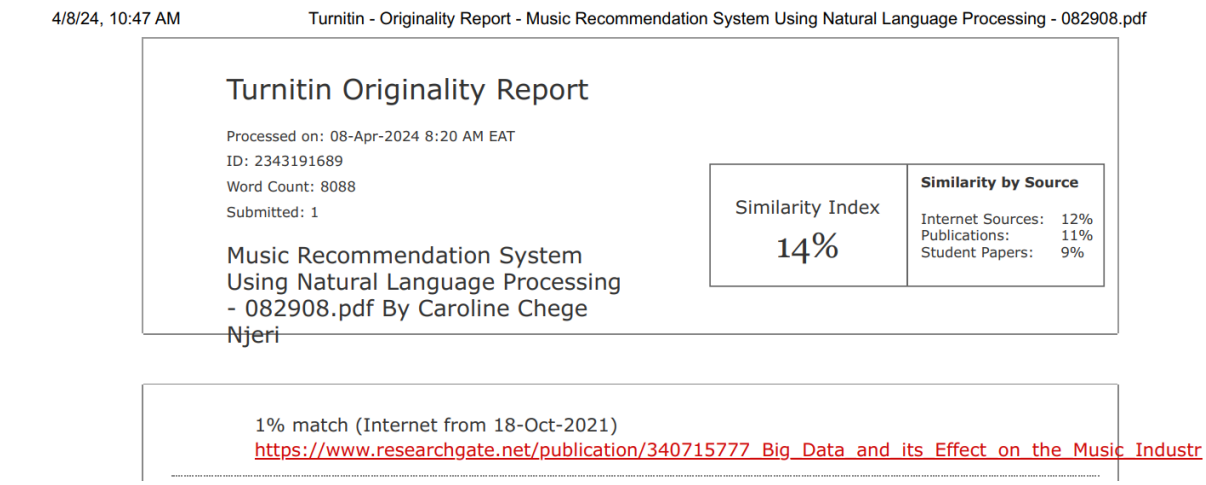
Welch, G. F., Biasutti, M., MacRitchie, J., & McPherson, G. E. (2020). The Impact of Music on Human Development and Well-Being. *Frontiers in Psychology, 11*. From https://www.frontiersin.org/articles/10.3389/fpsyg.2020.01246/full

Yue, & Xi. (2011). *The Music Industry in the Social Networking Era.*

Zehr, H. (2021). An Economic Analysis of the Effects of Streaming on the Music Industry in Response to Criticism from Taylor Swift. *Major themes in Economics, 23*. From https://scholarworks.uni.edu/cgi/viewcontent.cgi?article=1154&context=mtie

Appendices

Appendix A: Similarity Report



4/8/24, 10:47 AM Turnitin - Originality Report - Music Recommendation System Using Natural Language Processing - 082908.pdf

Turnitin Originality Report

Processed on: 08-Apr-2024 8:20 AM EAT
ID: 2343191689
Word Count: 8088
Submitted: 1

Music Recommendation System Using Natural Language Processing - 082908.pdf By Caroline Chege Njeri

Similarity Index	Similarity by Source	
14%	Internet Sources:	12%
	Publications:	11%
	Student Papers:	9%

1% match (Internet from 18-Oct-2021)
https://www.researchgate.net/publication/340715777_Big_Data_and_its_Effect_on_the_Music_Industr

Appendix B: Ethical Clearance Release Letter

Strathmore UNIVERSITY

9th April 2024

Caroline Chege

082908

Caroline.Chege714@strathmore.edu

Dear Caroline,

RE: Music Recommendation System Using Natural Language Processing

This is to inform you that the Office of Graduate Studies on 9th April 2024 received your request for intervention/assistance following the referral of your matter by the Strathmore University Institutional Scientific and Ethics Review Committee (SU-ISERC) to our Office due to the fact that you stated that you had already collected and/or analysed its data prior to seeking Ethical clearance. The ethics approval process is ONLY done before any collection of primary or secondary data.

We have taken note of your response that the information that your email admission and request for leniency.

This is a letter for you to proceed with the next steps of your academic requirements.

Please be advised, that in future, all research proposals should be submitted to the SU-ISERC through the RHInnO Ethics platform: https://strathmoreuniversity.rhinno.net/login

Disclaimer: *1) This is not in any way an ethical approval letter. 2) Should there be any legal implications/actions emanating from the research in terms of any ethical violations, you will be personally liable.*

Yours sincerely,

Dr. Bernard Shibwabo
Director of Graduate Studies

Ole Sangale Rd, Madaraka Estate. PO Box 59857-00200, Nairobi, Kenya. Tel +254 (0)703 034000
Email admissions@strathmore.edu www.strathmore.edu