DEVELOPMENT OF A NIGERIAN MUSIC COLLECTION AND RECOMMENDATION MODEL USING MACHINE LEARNING

BY

FALUYI TESTIMONY OLUWADUYILEMI

BU19CIT1066

A PROJECT SUBMITTED TO

COMPUTER SCIENCE PROGRAMME,

COLLEGE OF COMPUTING AND COMMUNICATION STUDIES,

BOWEN UNIVERSITY, IWO, OSUN STATE NIGERIA.

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

AWARD DEGREE OF BACHELOR OF SCIENCE (B.Sc.) IN

COMPUTER SCIENCE

JULY 2023

CERTIFICATION

This is to certify that this project, Development of a Nigerian music collection and recommendation model was carried out by FALUYI TESTIMONY OLUWADUYILEMI (Matriculation Number: BU19CIT1066) of the Computer Science Programme under the supervision of

....

Mr. A.I. Oyebade Date

(Supervisor)

....

Dr. D. O. Olanloye, Date

(Programme Coordinator)

DEDICATION

This project is dedicated to Almighty God, the one who saw me through the completion of this project and to my wonderful parents Dr. S. O. Faluyi and Dr. B. I. Faluyi, who have made it possible to get to this level in my academic pursuit. Also, I would like to dedicate this report to my lecturers and colleagues hoping that it would help increase our knowledge.

ACKNOWLEDGEMENTS

I would like to acknowledge the Almighty God for sustenance, wisdom and understanding. I would like to acknowledge the effort of my supervisors Mr. A.I. Oyebade for taking the time to guide me through this work.

With a deep sense of appreciation, respect and gratitude, I would like to acknowledge the efforts of the Provost, Prof. P. A. Idowu ,My gratitude goes to the Program Coordinator, Dr. D. O. Olanloye, alongside Dr. O.N. Emuoyibofarhe, Dr. C. Umezuruike, Dr. A.O. Akinwunmi, Dr. H. O. Aworinde, Dr. M. O. Oyelami, Dr. Adepegba, Mrs. Ife Awoniran, Mr. A. A. Adeyemo, Mr. B.P. Ayanniyi, Mrs. O.O. Olaniran, Mr. A.I. Oyebade, Mr. Falana and Mr. Alao for giving me a solid footing in the field of Computer Science and Information Technology; I say God bless you richly.

I appreciate my parents; Dr. S. O. Faluyi and Dr. B. I. Faluyi for their unending love and support. I cannot forget the contributions of my course mates and friends to the successful completion of this work. I am forever grateful. Thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY OF TERMS

KNN	K-Nearest Neighbour
CF	Collaborative Filtering
TF-IDF	Term Frequency-Inverse Document Frequency
SVM	Support Vector Machine
WCSS	Within Cluster Sum of Squares
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
FMA	Free Music Archive
CSV	Comma-Seperated Values
DBI	Davies-Bouldin Index
CHI	Calinski-Harabasc Index
ARI	Adjusted Rand Index
NMI	Normalized Mutual Information
GHZ	Gigahertz
GB	Gigabyte
RAM	Random Access Memory

ABSTRACT

This study presents a Nigerian music collection and recommendation model that utilizes clustering and the K-means algorithm. The model aims to enhance music discovery and provide personalized recommendations to Nigerian music enthusiasts. The dataset includes song metadata, artist information, genre labels, and user listening history. Pre-processing techniques are applied to handle missing values and extract relevant features. Clustering using K-means is performed to group songs and artists based on feature similarity. The model assigns users to the most relevant cluster, enabling targeted recommendations within their preferred music styles. The model's performance is evaluated using metrics such as silhouette score and user feedback, and iterative refinement is conducted to improve its accuracy and effectiveness. Overall, the model facilitates personalized music exploration and recommendation within the diverse landscape of Nigerian music.

CHAPTER ONE
INTRODUCTION

1.1 Background of the Study

With the explosion of network in the past decades, internet has become the major source of retrieving multimedia information such as video, books, and music etc. People have considered that music is an important aspect of their lives and they listen to music, an activity they engaged in frequently. However, the problem now is to organize and manage the millions of music titles produced by society. A good music recommender system should be able to automatically detect preferences and generate playlists accordingly. The proposed system is to detect music plagiarism based on music similarity. The plagiarism system extracts the music from input and finds music that are close to the query music which the query has plagiarized. Meanwhile, the development of recommender systems provides a great opportunity for industry to aggregate the users who are interested in music. We need to generate the best music recommendation system which is need to predict based on customization, by using KNN, Machine Learning (Verma et al., 2021).

Elaine Rich described her Grundy library system; It is used to recommend books to users following a short interview in which the user is initially asked to fill in his first and last name and then, in order to identify the users' preferences and classify the "stereotype", Grundy asks them to describe themselves in a few keywords. Once the information has been recorded, Grundy makes an initial suggestion by displaying a summary of the book. If the suggestion does not, please the user, Grundy asks questions to understand on which aspect of the book it has made a mistake and suggests a new one. However, its use remains limited and Rich faces problems of generalization (Elaine, 1979).

A music recommendation system is a system that learns from the users past listening history and recommends songs which they would probably like to hear in the future. By using a music recommend system, the music provider can predict and then offer the appropriate songs to their users based on the characteristics of digital music that has been heard previously (Verma et al.,2021).

Recommendation Systems are everywhere and pretty standard all over the web. Currently, there are many music streaming services, like Pandora, Spotify, etc., which are working on building high-precision commercial music recommendation systems. Amazon, Netflix, and many such companies are using Recommendation Systems.

1.2 Statement of the Problem

Music lovers have tremendous taste of identifying their type preference, this has resulted in listening to any song that comes their way, there is therefore need to develop a music recommender model for Nigerian music lovers that will satisfy their choice, hence this study.

1.3 Aim of the Study

The aim of the study is to develop a music recommendation model that is able to recommend music to a user based on the past listening history of a user.

1.4 Objectives of the Study

The objectives of the study are to:

1. collect music dataset.
2. develop a music recommender model for users
3. evaluate the model.

1.5 Methodology

The methodology must align with the steps in the objectives:

i. The music dataset will be collected from the Nigeria music collection. The dataset will be preprocessed with python preprocessing techniques.

ii. The preprocessed dataset will be used to formulate a music recommender system for Nigerian music lovers using KMeans algorithm.

iii. The model will be evaluated using silhouette score.

1.6 Significance of the Study

1. The proposed model enables the music provider to be able to predict and then offer the appropriate songs to their users.

2. The proposed model allow by use of property of song to recommend music for a user.

1.7 Scope of the Study

The project is focused on building a music recommendation system that recommends music based on the property of the song.

1.8 Structure of the Project

This project is divided into five sections. The first chapter covers the study's background, goals, and objectives, as well as the problem statement, suggested solution, and methodology. In order to completely appreciate how each technology works and the effect, Chapter Two offers a literature analysis of areas such as music recommendation model evaluation and other technologies involved. In Chapter Three, you'll learn how to analyze and build a model using various use case diagrams and other diagrams as needed. The model’s evaluation is covered in Chapter Four. The final chapter contains an overview of the entire project, as well as a conclusion and recommendations for further research.

CHAPTER TWO

LITERATURE REVIEW

2.1 Overview

The chapter started with the general discussion on music and the various types of music. The discussion majorly focus on the Nigerian music and the various types of music are used to. This chapter also gives detailed explanation of recommender system as a sub-field of text mining, various ways in which recommender system can be used to recommend good and services to the users. The various machine learning techniques as applied to recommender system were also discussed. The chapter concluded on the existing studies on recommender system.

2.2 Music

Music is very popular in modern life, and the amount of digital music increases rapidly nowadays. How to manage a large digital music database has arisen as a crucial problem. It is generally defined as the art of arranging sound to create some combination of form, harmony, melody, rhythm, or other expressive content. (Dan-Ning et al., 2002)

Definitions of music vary depending on culture, though it is an aspect of all human societies and a cultural universal. While scholars agree that music is defined by a few specific elements, there is no consensus on their precise definitions.

Although the topic itself spans into academic fields, criticism, philosophy, and psychology, the creation of music is typically split into three categories: musical composition, musical improvisation, and musical performance. A wide variety of instruments, including the human voice, can be used to play or improvise musical compositions.

2.2.1 Types of music

- Classical music

- Country music
- Electronic Dance music (EDM)
- Hip-hop
- Indie rock
- Jazz
- K-pop
- Metal
- Oldies
- Pop
- Rap
- Rhythm and Blues (R&B)
- Rock
- Techno

2.2.2 Nigerian music

There are many different types of folk and popular music in Nigeria. There isn't much documentation of the country's pre-European music history, but performers and their instruments have been depicted in bronze carvings from the 16th and 17th centuries. Indigenous, Apala, Aurrebbe Music, Rara Music, Were Music, Ogene, Fuji, Jùj, Afrobeat, Afrobeats, Igbo Highlife, Afro-juju, Waka, Igbo Rap, Gospel, and Yo-pop are the genres most well-known outside. The more than 250 ethnic groups in the nation, each with their unique methods, instruments, and melodies, are represented by several folk music genres. The Igbo, Hausa, and Yoruba are the three major ethnic groups. (“Music of Nigeria” 2023)

The current population of Nigeria is 221,476,147 as of Wednesday, July 5, 2023, based on Worldometer elaboration of the latest United Nations data. The total land area is 910,770 Km2 (351,650 sq. miles) (Worldometer, 2023)

2.2.3 Types of music in nigeria

- Afrobeat
- Juju music
- Highlife
- Apala
- Fuji
- Reggae
- Afrobeats/Hip hop
- Alternative (Alte)
- Classic
- Gospel

2.3 Overview of Recommender System

In this thesis the focus will be on recommendation systems. A recommendation system is a set of techniques and services whose purpose is to propose to users' articles that are likely to interest them. (Marine, 2020)

They are presently implemented on multimedia content distribution platforms (Netflix, Deezer, Spotify, TikTok, YouTube), online sales platforms (Amazon, eBay), social networks (Facebook, Twitter, Instagram). Recommender systems are particularly useful when the number of users and articles becomes very large. That is because users are unlikely to know all the richness of the catalogue offered by the service, and it can be argued that it is almost impossible to make a personalized human prescription for all the users of a service. The purpose of the recommendation system is to lead users through the vast amount of data available, particularly in e-commerce platforms, filtering this data to automatically propose to each consumer the items that are likely to be of interest to them. (Marine, 2020)

Taking these peculiarities into consideration it is now necessary to make an adequate recommendation. The more the user has confidence in the recommendation system and knows how it works, the more effective it will be. A successful recommendation involves a trade-off between exploitation and exploration. (Markus, 2017)

This implies that we need to find the appropriate balance between: novelty and familiarity, diversity and similarity as well as popularity and personalization. (Sean et al., 2006)

Finally, transparency with users is a crucial point. It has been proved that explaining how the algorithm works to the user improves their confidence and therefore the time they will spend on the platform, firstly to perfect their profile and secondly because they will get better recommendations. (Markus, 2017)

2.3.1 Types of recommender systems

There are three main recommendation systems which provide the ability to create music playlists adapted to a user: collaborative filtering, content-based information retrieval techniques, and context-based recommendation. A combination of the previous techniques is possible and is called hybrid. (Peter et al., 2016)

1. **Collaborative approach:** This recommendation method is based on the analysis of both the behaviour of the listeners and the behaviour of all others users of the platform. The fundamental assumption here is that the opinions of other users can be used to provide a reasonable prediction of another user's preferences for an item that they have not yet rated (Javier et al., 2018): a user is given recommendations based on users with whom they share the same tastes with. Indeed, during years, in order to choose music,

restaurants, movies, etc... We have been asking our friends, family, and colleagues to recommend something they liked. And it is this mechanism that is attempted to be reproduced here. Netflix was a pioneer of this method (based on stars given by other users) but it is now widely used, including for Spotify’s Discover Weekly. The first family of collaborative filtering methods is called memory-based approach (Javier et al,2018). The principle is to store all data in a User’s/Songs matrix.

The last step is to find similarity between vectors to be able to recommend music to listeners, to do so there are two methods:

• User-user similarity: comparing the listener vector with others user’s vectors to find those who have similar tastes. It is derived by contrasting the interactions or item rating habits of several consumers. To measure how similar people are to one another, a variety of similarity metrics can be applied; some of the more popular ones are as follows:

- The linear correlation between two users' ratings is measured by the Pearson correlation coefficient. A strong positive association is indicated by a Pearson correlation coefficient of 1, a strong negative correlation by a Pearson correlation coefficient of 1, and no correlation by a Pearson correlation coefficient of 0.
- Cosine similarity: It determines the cosine of the angle formed by the rating vectors of two different users. Between -1 and 1, where 1 denotes perfect similarity, -1 denotes perfect dissimilarity, and 0 denotes no similarity, is the cosine similarity scale.
- Similarity according to Jaccard: It determines the proportion of the intersection of ratings from two users to the union of ratings. The Jaccard similarity scale runs from 0 to 1, with 0 denoting no resemblance and 1 denoting total similarity.

• Item-item similarity: comparing tracks vectors to find which one is the closest to the actual listened music. Another strategy used in collaborative filtering to provide tailored recommendations is item-item similarity. Item-item similarity evaluates the similarity of items based on the ratings or interactions of users, as opposed to user-user similarity, which focuses on comparing user preferences.

The following steps are involved in calculating item-item similarity:

- Constructing the item-item similarity matrix: The degree of similarity between each pair of objects is calculated. It is possible to employ a variety of similarity metrics, including cosine similarity, Pearson correlation coefficient, and Jaccard similarity. An ordinary matrix is used to store the similarity values.
- Identifying similar items: The most similar items to a target item can be found once the item-item similarity matrix has been created. The most common method for doing this is by choosing the items with the highest similarity values.
- Generating recommendations: The ratings or preferences of users for comparable items can be used to generate recommendations. This can be accomplished using basic strategies like weighted averaging or more complex ones like matrix factorization.

Similarity between items has some advantages over similarity between users. Since the similarity matrix is frequently smaller than the user-item matrix, it is especially helpful when working with huge user bases. Since human preferences can change over time while item qualities are generally fixed, item-item similarity is also more stable over time than user-user similarity.

Techniques like neighborhood-based algorithms or matrix factorization can be used to increase the scalability of item-item similarity calculations. By simplifying the math involved, these techniques increase the effectiveness of providing recommendations based on item-item similarity.

There is a second approach called model-based: the goal is to predict the user’s rating for missing items using machine learning models. The key advantage of the collaborative approach is that we do not need to analyze and extract features from the raw files, so there is no need to have the audio files, nor to have an in-depth knowledge of music or physics (Marine, 2020). Moreover, it brings serendipity, it is the effect of surprise that the user can receive by being given a relevant recommendation that they would not have found alone. There are three major drawbacks according to Marine Chemeque-Rabel:

- **Cold start issues**: it designates two issues: new user problem and new item problem. The former reflects the lack of user data to make a relevant recommendation while the latter reflects the fact that we do not know who to recommend new items to. This refers to the difficulties that come up when attempting to make precise suggestions for new products or new users who have scant or no historical data at their disposal. Because collaborative filtering relies on prior data to spot trends and make recommendations, these cold start conditions provide a challenge.
- **Scalability issues**: This another issue with the collaborative approach where a large number of users and items requires high computing resources. Collaborative filtering has a substantial scaling difficulty, particularly when working with huge user bases and item collections. The computational complexity and memory requirements of conventional collaborative filtering methods drastically rise as the

number of users and objects rises. In real-world systems, this may result in performance deterioration and practical restrictions.

- **Sparsity issues**: This is another bottleneck identified with collaborative approach because the number of items is large, one user can only rate a small subset of them. In collaborative filtering, sparsity is a common problem since most elements in the user-item interaction matrix are either missing or empty. Users have only rated or engaged with a small portion of the accessible objects, which leads to this sparsity. The sparsity problem can have a substantial impact on how well collaborative filtering algorithms work and produce unsatisfactory recommendations.

2. **Memory-Based Collaborative Filtering (Neighbourhood based):** People with similar interests are combined to form a group and every user is a part of that. User –based CF and Item implement and scales well with correlated items. There is no need for items being recommended. There are many limitations of memory problem, sparsity and their dependencies on human ratings. One kind of collaborative filtering algorithm directly bases suggestions on the user-item interaction data is memory-based collaborative filtering. It uses similarities among users or objects to find patterns and produce tailored recommendations. User-based and item-based are the two basic strategies used in memory-based collaborative filtering.

   - **User-based collaborative filtering**:

     - Collaborative filtering that is based on user ratings or interactions aims to connect people that are similar to one another. It determines which people are most similar to a target user by calculating user similarity between them.

    - Following the discovery of related users, suggestions are created by combining their ratings or interests. The presumption is that consumers who have previously rated products similarly will probably continue to have similar preferences.
    - Collaborative filtering based on user input is simple to use and put into practice. However, as the similarity must be determined for each user pair, it might be computationally expensive when working with a large number of users.
- **Item-based collaborative filtering**:
    - The goal of item-based collaborative filtering is to identify related items based on user interactions or rating trends. It determines which items are most similar to a target item by calculating the degree of similarity between them.
    - Recommendations are created once similar things have been found based on user ratings or interactions with those similar items. The presumption is that users who have previously expressed interest in similar things would probably continue to do so in the future.
    - Given that things rather than people are used in the similarity calculations, item-based collaborative filtering is computationally efficient. In comparison to user similarity, item similarity also has a tendency to be more consistent over time.

Memory-based cooperative filtering offers the following benefits:

- Simplicity: Implementing and comprehending memory-based collaborative filtering is not too difficult.

- Memory-based methods that take use of user or item similarities can turn up unexpected and fortuitous recommendations.
- Interpretability: Memory-based methods offer transparency because they base recommendations on users' or objects' explicit similarities.

Memory-based collaborative filtering, however, also has significant drawbacks:

- Scalability: The computational complexity and memory needs of memory-based approaches might become prohibitive as the quantity of users or items increases.
- Memory-based approaches are susceptible to data sparsity since it can lead to limited or erroneous suggestions in the case of missing or sparse data.
- In situations involving new users or things, when there is insufficient evidence to determine meaningful similarities, memory-based approaches have difficulty.

3. **Content-based approach:** The content-based recommendation consists of the analysis of the content of the item's candidates for recommendation. This approach aims to infer the user's preferences in order to recommend items that are similar in content to items they have previously liked. This method does not need any feedback of the listener, it is only based on sound similarity which is deduced from the features extracted from the previous listened songs (Javier, 2018). This method is based on the similarities between the different items. To estimate similarities, it is a matter of extracting features to best describe the music. The Machine Learning algorithms then recommends the closest item to those that the user already likes.

Cosine similarity (or similarities) is a popular way to gauge how similar two objects or documents are based on the aspects of their content when using a content-based approach.

In order to provide suggestions, content-based filtering relies on examining the traits or qualities of things and identifying commonalities among them. The common application of cosine similarity in a content-based strategy is as follows:

- Items are represented by feature vectors that include information about their content. These features may come from a variety of sources, including text analysis (word embeddings, TF-IDF, and keywords), image analysis (visual features), and audio analysis (audio features). A dimension in the feature vector is represented by each feature.
- Vector normalization: It is typical to normalize the feature vectors to have a unit length prior to determining cosine similarity. This step makes sure that the feature values' magnitude or scale have no bearing on the similarity calculation.
- Cosine similarity computation: The formula below is used to calculate the cosine similarity between two item feature vectors:

  cosine_similarity(A, B) = (A . B) / (||A|| * ||B||)

  - The feature vectors of two items are represented by A and B.
  - A.B stands for the vectors A and B's dot product.
  - The Euclidean norms (magnitudes) of the vectors A and B are represented by ||A|| and ||B||, respectively.

  The resulting cosine similarity value falls between -1 and 1, with 1 denoting perfect similarity, -1 denoting perfect dissimilarity, and 0 denoting no similarity.

Due to its advantageous characteristics, cosine similarity is frequently used in content-based approaches.

It is, therefore, necessary to create items profiles based on features extracted from items. Moreover, this method requires user profiles based on both their preferences and their history on the platform. These profiles will be in the following form: a list of weights (which reveals the importance) corresponding to each feature we have selected. The main advantage of this approach is that an unknown music is just as likely to be recommended as a currently popular one, or even a timeless one. This allows new artists with a few" views" to be brought up as well. Moreover, the problem of the cold start and in particular of the new items is thus avoided: when new items are introduced into the system, they can be recommended directly, without requiring integration time as is the case for recommendation systems based on a collaborative filtering approach. The negative point is that this method limits the diversity in the recommendation, it tends to over-specialize. Moreover, the integration of a new user cannot be instantaneous, they have to listen and evaluate a certain number of songs before being able to receive recommendations, this is the user cold start.

4. **Context-based approach:** Studies have shown that the mood, activity, or even the location of the person influences the music they want to listen to. We listen to music in a given moment, in a predefined emotional state, and established circumstances (party, work). And these predispositions will play a decisive role in the way we feel about the music. Although there are many applications of this type of recommendations such as tourist guide applications with adaptive ambient songs, there are not many concrete applications on this subject. Many barriers still block the research on this field. Indeed, the nature of the data to be taken into account is highly varied and depends on the environment or the user themselves. An even more significant issue is the lack of data available for research purposes. In the real world it is not easy to retrieve them either, as users do not always want to transmit as much information from their mobile phone sensors.

5. **Hybrid approach:** It is also possible to combine the previous complementary methods to create a recommendation system called hybrid. It can also be based on all other lesser-known methods such as location-based recommendation. This method can alleviate the problems of cold start and sparsity. Several implementations can be set up, first of all the recommendation systems can be mixed into one. It is also possible to keep several systems separated and assign them weights, or the ability to switch between systems at will. Finally, it is possible to extract results from one system, to then be used as an input for the next one.

Table 2.1: Comparison of different recommendation system

Model	Description	Advantages	Disadvantages
Collaborative filtering	Based on user or item similarities, analyses user activity and preferences to make music recommendations.	Simple to use and able to offer reliable advice	Cold-start issue with new users or objects
Content-based filtering	Provides music recommendations based on the music's own qualities (genre, tempo, lyrics, etc.).	Useful for new or specialized products; independent of user data	Lack of sophisticated user preferences and possible diversity
Hybrid Models	Combines collaborative filtering and content-based filtering methods to produce suggestions that are more accurate.	Incorporates the advantages of both strategies	More difficult to implement and requiring more computing power

2.3.2 Applications of recommender systems

These are various ways in which a recommender system can be applied:

- **E-commerce**: In order to provide consumers with individualized product recommendations based on their browsing history, purchasing patterns, and preferences, recommender systems are widely employed in online retail platforms. These technologies increase revenue, enhance consumer satisfaction, and offer individualized shopping experiences.
- **Streaming services**: Recommender systems are used by platforms like Netflix, Spotify, and YouTube to make appropriate movie, TV show, song, and video suggestions to their viewers. These systems offer personalized content recommendations by examining user preferences, viewing behavior, and ratings, increasing user engagement and satisfaction.
- **Social media**: Recommender systems are used by social media sites to suggest to users' relevant posts, material, and profiles. These systems assist users in finding new information and establishing connections with like-minded people by taking into account elements including user interests, social connections, and engagement habits.
- **News and Content Aggregation**: News apps and content aggregators employ recommender systems to make customized suggestions for news items, blog entries, and other content to users based on their interests and reading habits. This aids consumers in staying informed and finding pertinent information amid a sea of data.
- **Travel and Hospitality**: Recommender systems help consumers identify appropriate hotels, airlines, and vacation packages by taking their tastes, finances, and past travel experiences into account. Based on user preferences and location, these systems can also suggest dining establishments, activities, and tourism destinations.

- **Job Portals**: Job search platforms match job seekers with relevant job opportunities based on their skills, qualifications, and preferences by using recommender algorithms. These tools increase the effectiveness of the job search process and direct consumers to relevant employment options.
- Online learning platforms, marketplaces, and email providers are just a few examples of services that use recommender systems to customize user experiences. These systems recommend pertinent emails, courses, products, and services based on user behavior analysis to increase user engagement and satisfaction.
- **Targeted advertising**: Recommender systems are essential for targeted advertising because they provide users with relevant ads based on their demographics, interests, and online activity. These programs aid marketers in campaign optimization, click-through rate enhancement, and conversion rate enhancement.

2.3.3 Music recommender system

The recommendation systems are more and more used in many fields: hotels, travels, products. But the musical field has some particularities to take into account. The first factor to consider is the duration of a music track. As a track is short, it is less critical to make a bad recommendation than it is for a movie or a book, for example. The user can also quickly browse through the music to quickly see if it suits their taste or not. A second specificity is the number of tracks available, indeed the choice is very wide, it is estimated that at least tens of millions of songs are accessible on the Internet. It is common for repeated recommendations of the same music to be appreciated. While for trips or movies the user is looking for diversity, the user may like to listen to the same music over and over again. Moreover, it is possible that at the first listening the user was not attentive since listening to music is often done in parallel with another activity (sport, work). Attentive listening requires quality hardware, the proper mood, and exclusive attention time. Moreover, it is quite easy to extract a set of features from one piece of music. Indeed, information can be extracted through signal processing, thanks to musical knowledge, thanks to lyrics, or just using user feedback. Old music is as relevant as new music: recent music, as well as music from a few decades ago or classical music can be as enjoyable. It is a matter of correctly understanding the user’s tastes. It must also be taken into account that music listening is often passive: the listener doesn’t necessarily listen attentively to it: in shops, bars, while working. The last point that distinguishes music from other items that may be recommended is that music is often played in sequence. Indeed, as they are short, they are often chained together in the form of a playlist.

On the first hand, exploitation consists in playing safe music, music that the recommender knows the user likes. It’s called lean-back experience and it brings short-term rewards. On the other hand, exploration is about playing new music, making new discoveries. It’s called lean-in experience and it brings long-term rewards. If it’s properly gauged, a little serendipity may please.

A relevant recommendation must also reflect the listener context. It cannot be only based on music and listener properties, it is needed to take into account the mood, the activity. For example, someone who is working does not want to listen to the same music as when they are running.

2.4 Machine Learning

Over the past decade, Machine Learning has become one of the main stays of information technology. Machine learning is used to teach machines how to handle the data more efficiently Sometimes, after viewing the data, we cannot interpret the pattern or information from data, thus, the use of machine learning.

The machine learning algorithms find the patterns in the training dataset, which is used to approximate the target function and is responsible for mapping the inputs to the outputs from the available dataset. These machine learning methods depend upon the type of task and are classified as Classification models, Regression models, Clustering, Dimensionality Reductions, Principal Component Analysis, etc.

Machine learning is no exception, and a good flow of organized, varied data is required for a robust machine learning solution. In today’s online-first world, companies have access to a large amount of data about their customers, usually in the millions. This data, which is both large in the number of data points and the number of fields, is known as big data due to the sheer amount of information it holds.

2.4.1 Supervised machine learning

Supervised learning algorithms are used when the output is classified or labeled. These algorithms learn from the past data that is inputted, called training data, runs its analysis and uses this analysis to predict future events of any new data within the known classifications. The accurate prediction of test data requires large data to have a sufficient understanding of the patterns. The algorithm can be trained further by comparing the training outputs to actual ones and using the errors for modification of the algorithms. In supervised learning, the ML algorithm is given a small training dataset to work with. This training dataset is a smaller part of the bigger dataset and serves to give the algorithm a basic idea of the problem, solution, and data points to be dealt with. The training dataset is also very similar to the final dataset in its characteristics and provides the algorithm with the labeled parameters required for the problem.

The algorithm then finds relationships between the parameters given, essentially establishing a cause-and-effect relationship between the variables in the dataset. At the end of the training, the algorithm has an idea of how the data works and the relationship between the input and the output. This solution is then deployed for use with the final dataset, which it learns from in the same way as the training dataset. This means that supervised machine learning algorithms will continue to improve even after being deployed, discovering new patterns and relationships as it trains itself on new data.

Supervised machine learning can be categorized into several types based on the prediction problem and the algorithms used:

- **Regression**: Continuous numerical values are predicted using regression methods. To build a regression model, they examine the relationship between input variables and a continuous target variable. Regression techniques are:
    - **Linear Regression**: This is a statical regression method which is used for predictive analysis. It is one of the very simple and easy algorithms which works on regression and shows the relationship between the continuous variables.

      Mathematical Equation:

      $Y = aX + b$

      Y = Dependent Variable (Target Variable)

      X= Independent Variable (Predictor Variable)

      a and b are linear coefficient.

      Application of linear regression:

      Analyzing trends and sales estimates.

      Salary forecasting

      Real estate prediction
- **Random Forest**: Random Forest is one of the most powerful supervised learning algorithms which is capable of performing regression as well as classification tasks.

  The Random Forest regression is an ensemble learning method which combines multiple decision trees and predicts the final output based on the average of each tree output. The combined decision trees are called as base models, and it can be represented more formally as:

$g(x) = f_0(x) + f_1(x) + f_2(x) + ...$

- **Classification**: When categorizing data instances into preset classes or categories is required to solve a prediction problem, classification methods are used. They create a classification model by learning from labeled training data. Logistic regression, Naive Bayes, Decision Trees, Random Forests, Support Vector Machines (SVM), and Neural Networks are a few examples of categorization methods.

  In classification algorithm, a discrete output function(y) is mapped to input variable(x).

  y=f(x), where y = categorical output.

  The best example of an ML classification algorithm is Email Spam Detector.

- **Support Vector Machines (SVM):** SVM is a flexible method that may be applied to both regression and classification tasks. It develops a hyperplane or collection of hyperplanes that effectively classify data points into groups or forecast a continuous output value.

2.4.2 Unsupervised machine learning

Unsupervised learning algorithms are used when we are unaware of the final outputs, and the classification or labeled outputs are not at our disposal. These algorithms study and generate a function to describe completely hidden and unlabelled patterns. Hence, there is no correct output, but it studies the data to give out unknown structures in unlabelled data. In supervised learning, the labels allow the algorithm to find the exact nature of the relationship between any two data points. However, unsupervised learning does not have labels to work off of, resulting in the creation of hidden structures. Relationships between data points are perceived by the algorithm in an abstract manner, with no input required from human beings. The creation of these hidden structures is what makes unsupervised learning algorithms versatile. Instead of a defined and set problem statement, unsupervised learning algorithms can adapt to the data by dynamically changing hidden structures. This offers more post-deployment development than supervised learning algorithms.

The unsupervised learning algorithm can be further categorized into two types of problems:

Clustering: Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group. Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.

The clustering technique is commonly used for statistical data analysis.

- **K-Means Algorithm**: The k-means algorithm is one of the most popular clustering algorithms. It classifies the dataset by dividing the samples into different clusters of equal variances. The number of clusters must be specified in this algorithm. It is fast with fewer computations required, with the linear complexity of O(n).

  The performance of the K-means clustering algorithm depends upon highly efficient clusters that it forms. But choosing the optimal number of clusters is a big task. There are some different ways to find the optimal number of clusters, but here we are discussing the most appropriate method to find the number of clusters or value of K. The method is given below:
- **Elbow method**: The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS value. WCSS stands for Within Cluster Sum of Squares, which defines the total variations within a cluster. The formula to calculate the value of WCSS (for 3 clusters) is given below:

  WCSS = $\sum_{p_i\ in\ cluster1} distance(p_i C_1)^2 + \sum_{p_i in\ cluster\ 2} distance(p_i C_2)^2 + \sum_{p_i in\ cluster\ 3} distance(p_i C_3)^2$

In the above formula of WCSS, $\sum_{p_i in\ cluster\ 1} distance(p_i C_1)^2$: It is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the other two terms.

- **Association**: An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis.

2.4.3 Semi-supervised machine learning

This type of machine learning algorithm uses the trial-and-error method to churn out output based on the highest efficiency of the function. The output is compared to find out errors and feedback, which are fed back to the system to improve or maximize its performance. The model is provided with rewards which are basically feedback and punishments in its operations while performing a particular goal.

Based on the psychological concept of conditioning, reinforcement learning works by putting the algorithm in a work environment with an interpreter and a reward system. In every iteration of the algorithm, the output result is given to the interpreter, which decides whether the outcome is favorable or not. In case of the program finding the correct solution, the interpreter reinforces the solution by providing a reward to the algorithm. If the outcome is not favorable, the algorithm is forced to reiterate until it finds a better result. In most cases, the reward system is directly tied to the effectiveness of the result. In typical reinforcement learning use-cases, such as finding the shortest route between two points on a map, the solution is not an absolute value. Instead, it takes on a score of effectiveness, expressed in a percentage value. The higher this percentage value is, the more reward is given to the algorithm. Thus, the program is trained to give the best possible solution for the best possible reward.

2.4.4 Reinforcement learning

These algorithms normally undertake labeled and unlabeled data, where the unlabelled data amount is large as compared to labeled data. As it works with both and in between supervised and unsupervised learning algorithms, therefore is called semi- supervised machine learning. Systems using these models are seen to have improved learning accuracy.

2.5 Empirical review

Yifan Hu et al., (2008): “A Fast Incremental Matrix Factorization Approach”

- Strengths: The paper introduces an efficient incremental matrix factorization approach for collaborative filtering, which allows for real-time updates to the recommendation model. The method is scalable and performs well on large datasets.
- Limitations: The focus is primarily on collaborative filtering, and the paper does not explore other techniques like content-based or hybrid methods. Additionally, the paper does not extensively address the challenges of the cold start problem or the sparsity of user-item interaction data.

Meinard Müller et al., (2005): “Content-Based Music Recommendation Using Audio Signal Analysis”

- Strengths: The paper introduces a content-based recommendation approach that analyzes audio signal features to measure music track similarity. It leverages information directly from the music itself, making it useful for recommending tracks with similar audio characteristics.
- Limitations: The approach relies heavily on the availability and quality of audio signal features. It may struggle with tracks that have sparse or incomplete audio features.

Additionally, this method does not capture user preferences explicitly, which limits its ability to provide personalized recommendations.

Keunwoo Choi et al. (2018): “Deep Learning for Music Recommendation: Challenges and Opportunities”

- Strengths: The paper explores the application of deep learning techniques for music recommendation, which have shown promising results in capturing complex patterns in music data. Deep learning models can effectively learn from large-scale data and handle sequential and temporal dependencies in music.
- Limitations: Deep learning models often require large amounts of data and significant computational resources for training. They can also be prone to overfitting if not properly regularized. Interpretability of deep learning models can be challenging, making it difficult to understand and explain the recommendations they provide.

Michael D. Ekstrand et al. (2011): "Music Recommendation and Discovery in the Long Tail"

- Strengths: The paper addresses the challenge of recommending music from the long tail, where less popular items have limited user data. It explores techniques like item-based collaborative filtering and personalized ranking to improve recommendations for niche or less-known tracks.
- Limitations: The methods discussed in the paper may still suffer from the sparsity of user-item interaction data, particularly for long-tail items. Additionally, the paper does not extensively cover content-based or hybrid approaches that can complement collaborative filtering in long-tail recommendation scenarios.

Claudio Gentile et al. (2011): "Exploiting Temporal Dynamics in Music Preferences for Recommendation"

- Strengths: The paper emphasizes the importance of temporal dynamics in music recommendation and explores methods to model users' changing preferences over time. It addresses the challenge of capturing evolving user tastes and incorporates time-decayed weighting and sequential pattern mining techniques.
- Limitations: The paper mainly focuses on user-based collaborative filtering and does not extensively cover other recommendation techniques. It may not fully capture the context and reasons behind users' changing preferences, which could limit the effectiveness of the proposed approaches.

Xavier Serra et al. (2008): "Hybrid Music Recommendation by Fusion of Content and Sequential User Behavior"

- Strengths: The paper proposes a hybrid recommendation system that combines content-based and collaborative filtering approaches. By leveraging both audio signal features and user behavior, the method can provide a more comprehensive and accurate recommendation.
- Limitations: The paper does not explicitly address the challenges of scalability or the cold start problem in hybrid recommendation systems. Additionally, the complexity of combining different techniques can introduce challenges in terms of model integration and parameter tuning.

Preeti Bajaj et al., (2020): "A Survey of Music Recommendation Systems and Future Perspectives"

- Strengths: The survey provides a comprehensive overview of various music recommendation techniques, covering collaborative filtering, content-based filtering, hybrid methods, and context-aware approaches. It offers a broad perspective on the field and highlights the advancements made in music recommendation systems.

- Limitations: The survey may not cover the most recent developments in the field, as the publication date is 2020. Additionally, the future perspectives provided may not account for more recent trends or emerging technologies in music recommendation.

Xinshi Chen et al. (2018): "A Survey on Deep Learning in Music Recommendation"

- Strengths: The survey specifically focuses on deep learning approaches for music recommendation, offering a detailed exploration of various deep learning models used in the field. It provides insights into the potential of deep learning techniques for capturing complex patterns in music data.
- Limitations: Since the publication date is 2018, some of the newer advancements and developments in deep learning-based music recommendation may not be covered. Additionally, the survey may not provide a comprehensive comparison of deep learning models against other traditional recommendation techniques.

Lianhong Cai et al. (2011): "A Comprehensive Survey of Music Recommendation Systems Based on Content Analysis and Collaborative Filtering"

- Strengths: The survey covers both content-based and collaborative filtering techniques for music recommendation, offering a comprehensive understanding of these approaches. It explores various content analysis methods and discusses their applications in music recommendation.
- Limitations: Since the publication date is 2011, the survey may not include the latest advancements in content-based and collaborative filtering techniques. The survey also does not extensively cover more recent approaches like deep learning or hybrid methods.

Bogdan Ionescu et al. (2014): "Music Recommendation Systems: Challenges and Opportunities"

- Strengths: The paper highlights the challenges in music recommendation systems, particularly in incorporating contextual information, diversity, novelty, and serendipity. It discusses the importance of considering these factors to improve the user experience in music recommendations.
- Limitations: The paper was published in 2014, so it may not cover the most recent advancements and trends in music recommendation systems. Additionally, while it presents challenges and opportunities, it may not provide specific solutions or techniques to address those challenges.

Li Chen et al. (2012): "Personalized Music Recommendation: A Survey"

- Strengths: The survey provides an overview of personalized music recommendation techniques, covering collaborative filtering, content-based filtering, hybrid methods, and knowledge-based systems. It explores the strengths and limitations of these approaches in delivering personalized recommendations.
- Limitations: As the publication date is 2012, the survey may not include the latest advancements in personalized music recommendation. It may not extensively cover newer approaches like deep learning or address recent challenges in the field.

Table 2.2 Empirical Review

Paper title	Author and Year of Publication	Description
Collaborative Filtering for Music Recommendation: A Fast Incremental Matrix Factorization Approach	Yifan Hu, Yedua Koren, and Chris Volinsky. 2008	In order to facilitate collaborative filtering in music recommendation systems, this work introduces the matrix factorization technique. When new user preferences or musical pieces are added, the authors' incremental technique enables quick modifications to the model.
Content-Based Music Recommendation Using Audio Signal Analysis	Meinard Müller and Michael Clausen. 2005	The authors suggest an audio signal feature analysis-based approach for matching music recordings based on their content. They gather pertinent data for recommendations using methods including tempo analysis, harmonic

		progression, and spectral properties
Deep Learning for Music Recommendation: Challenges and Opportunities	Keunwoo Choi et al. 2018	The use of deep learning algorithms for music recommendation is explored in this research. The difficulties of modeling musical data are discussed, including capturing temporal connections and managing sparse data. In the context of music recommendation, the authors introduce various neural network topologies and go over their advantages and disadvantages.
Music Recommendation and Discovery in the Long Tail	Michael D. Ekstrand et al. 2011	In the long tail, where fewer well-known or specialized items have scant user data, the research focuses on music suggestion. It examines methods for

		promoting music from the long tail, including item-based collaborative filtering, personalized rating, and context-aware recommendation.
Exploiting Temporal Dynamics in Music Preferences for Recommendation	Claudio Gentile et al. 2011	The significance of temporal dynamics in music suggestion is examined in this essay. In order to increase the precision of recommendations, it investigates techniques to simulate users' altering preferences over time, such as time-decayed weighting, session-based recommendation, and sequential pattern mining.
Hybrid Music Recommendation by Fusion of Content and Sequential User Behaviour	Xavier Serra et al. 2008	The authors suggest a hybrid recommendation system that incorporates techniques to collaborative and content-based filtering. To improve

		the suggestion process, they combine audio signal characteristics with sequential user activity, such as listening history and ratings.
A Survey of Music Recommendation Systems and Future Perspectives	Preeti Bajaj and Ugrasen Suman. 2020	An overview of several music recommendation systems, including collaborative filtering, content-based filtering, hybrid techniques, and context-aware methods, is given in this survey. It covers each technique's advantages and disadvantages and outlines potential future research trajectories.
A Survey on Deep Learning in Music Recommendation	Xinshi Chen et al. 2018	Particularly, deep learning methods for music recommendation are the subject of this survey. Convolutional neural

		networks (CNNs), recurrent neural networks (RNNs), and hybrid architectures are only a few examples of the deep learning models that are covered in this introduction. It talks about the opportunities and difficulties of using deep learning for music recommendation.
A Comprehensive Survey of Music Recommendation Systems Based on Content Analysis and Collaborative Filtering	Lianhong Cai et al. 2011	Both content-based and cooperative filtering methods for music recommendation are covered in this survey. It explores several content analysis techniques, such as audio signal analysis, analysis based on lyrics, and analysis based on metadata. Additionally, it looks into user-based, item-based, and model-based approaches to collaborative filtering.

Music Recommendation Systems: Challenges and Opportunities	Bogdan Ionescu et al. 2014	In this study, opportunities and limitations in music recommendation systems are briefly discussed. In order to increase the accuracy of recommendations, it addresses the value of contextual information, such as user context and social context. The difficulties of including originality, diversity, and serendipity in musical recommendations are also covered.
Personalized Music Recommendation: A Survey	Li Chen et al. 2012	This survey gives a general overview of methods for making tailored music recommendations. It encompasses a variety of strategies, including knowledge-based systems, collaborative filtering, content-based filtering, and

		hybrid techniques. It covers the main difficulties in recommending music to individuals and looks at potential future research areas

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.1 Overview

This chapter involves the study of different existing music recommendation models, so as to detect and highlight the drawbacks and limitations in those systems and come up with a solution for those drawbacks. A music recommendation model was developed with a high accuracy to eliminate the problem of music lover listening to song that's not preferred by them.

3.2 Proposed Model

The proposed music recommendation model will make use of K-Means clustering algorithm, an unsupervised machine learning algorithm to evaluate the music data. A K-Means clustering algorithm is a subset of unsupervised machine learning. K-Means is used because the output of the algorithm is unknown.

The music dataset will be gotten from Kaggle which is the biggest library for data for various fields. After the data collection, the data would be processed by removing undesired attributes. The K-Means model would be created and then the model would be trained and compiled. Performance metrics such as silhouette score and clusters graph would be our result.

3.2.1 Analysis of existing system

The current music recommendation model makes use of latent-factor methods which relies on user's choices to recommend music. The recorded data of the user listening to music is decomposed into a user preference matrix and a music feature matrix. By fitting the user's choices rather than scoring behaviors, the probability of observable user behaviour can be maximized rather than being tailored to a particular evaluation value. In order to effectively alleviate the problem of excessive data volume, this paper proposes a clustering algorithm.

The user and music dimensions are clustered separately, and the user interest preference matrix in each project cluster is used to find the neighbor user of the user corresponding to the item to be evaluated. Finally, a user-based collaborative filtering recommendation algorithm is applied to each user class cluster for recommendation.

3.2.2 Framework of existing system

The framework of a user-based collaborative filtering recommendation for music recommendation typically consists of several components which start at the data acquisition stage and ends at the evaluation stage. Here is a high-level overview of the key components involved:

1. **Data Acquisition:** The GTZAN and FMA dataset were used.

   GTZAN is a minimalist dataset containing one thousand music samples of 30 seconds from 10 different genres. Raw audio singly labelled are provided. (Bob, 2013)
   The FMA (Free Music Archive) is a large-scale dataset created for music analysis that is composed of more than 100,000 thousand tracks from 161 genres. It provides various track-level (including low-level features), artist-level and album-level metadata in csv files. (Michael et al., 2016)

2. **Data Pre-processing:** The input data is preprocessed to prepare it for analysis.

   For FMA dataset, under sampling and oversampling methods are used to solve class imbalance.

3. **Feature Extraction:** The music is in mp3 format for the FMA dataset and in wav format for GTZAN. The first step is to extract the spectral envelopes. The datasets were then split into a training set containing 90% of the data and a test set of 10%.

   From the spectral envelopes of each music, an attempt is made to extract as many useful features as possible in order to best describe all the characteristics of the music.

Depending on the features, there are three different extraction modes: either at a given time t, over the whole music, or (in most cases) by frame.

4. **Feature Selection:** To avoid information redundancy and limit parasitic features, the wrapper model.

The wrapper method was introduced by Kohavi and John (Ron et al., 1997). In this case, the evaluation is done using a classifier that estimates the relevance of a given subset of features. That is why the subset of features selected by this method matches the classification algorithm used, but the subset is not necessarily valid if the classifier is changed.

Most common implementations of wrapper are:

• Forward selection: start with no features and add the most relevant one at each step:

1. Choose the significance level (e.g. 0.05)
2. Select the feature that fits the model with the lower p-value
3. If p value < α, add the feature to the feature set and go back to step 2, else stop the process. α is the significance level.

• Backward elimination: start with every feature, and remove the insignificant one at each step:

1. Choose the significance level (e.g. 0.05)
2. Fit the model with all features in the feature set
3. Consider the feature with highest p-value
4. If p value > α, remove the feature from the feature set and go back to step 2, else stop the process.

• Stepwise Selection / Bidirectional elimination: Similar to forward selection, a feature is added to each iteration, but it also verifies the significance of features already added and can remove a feature through backward elimination if needed.

1. Choose the significance level (e.g. 0.05)
2. Perform steps 2 and 3 of forward selection
3. Perform steps 2, 3 and 4 of backward elimination
4. Repeat 2 and 3 until finding the optimal feature set

This method is considered the best. Indeed, it selects a small subset of features that are efficient with the classifier used, however there are two main drawbacks that limit these methods: firstly, the computation time is much longer than for the previous method, and the cross-validation often used to reduce the risks of overfitting increases this problem. The second challenge is to apply this selection mechanism for each classifier to test.

3.2.3 Drawbacks of existing system

While user-based collaborative filtering can achieve reasonable accuracy for music recommendation, they also have limitation.

First of all, it should be pointed out that the way our system is evaluated is controversial. Indeed, basing on metrics such as accuracy provides little information about the performance of our recommender. But the article quoted above specifies that recall, accuracy or confusion matrices do not really provide a true assessment of the ability of our system to recognize a genre. Moreover, the interpretations made from the results are based on human feelings. We base our interpretations on the instruments we recognize, on the way the music is sung or rapped, whereas current features are not able to focus on such information. (Bob, 2013)

The GTZAN dataset contains some errors in the labels. As the concept of genre may be ambiguous, a more in-depth analysis of the genres in FMA and GTZAN should also be done in order to determine whether music classified as” blues” in FMA for example would also be classified as” blues” in GTZAN.

This is why the interpretations made in this thesis are based on two assumptions. The first one is that the dataset created is coherent and consistent at the label level. The second is that the recommendation system uses cues (like number and type of instruments, music for dancing, relaxing) similar to those that would be used by a human trying to classify music. (Bob, 2013)

3.2.4 Framework of proposed model

This is a framework you use to create a clustering model that comprises of various steps. A generic framework for developing and accessing a clustering model is provided below:

Describe the issue: Clearly state the issue that clustering is intended to address. Establish the analysis's aims and objectives. Decide which characteristics or variables will be used for clustering.

Preparing the data for clustering through data processing: Handling missing numbers, dealing with outliers, and, if necessary, processing the data are all part of this step. To make sure that all variables have a comparable scale, you may also take into account feature scaling or normalization.

Selecting a clustering algorithm: The k-means algorithm is one of the most popular clustering algorithms. It classifies the dataset by dividing the samples into different clusters of equal variances. The number of clusters must be specified in this algorithm. It is fast with fewer computations required, with the linear complexity of O(n).

Set the hyperparameters: If the chosen clustering algorithm has hyperparameters (e.g., the number of clusters for k-means), specify the values for these hyperparameters. You may need to experiment or use techniques like cross-validation to find the optimal values.

Fit the model: Apply the clustering algorithm to the preprocessed data. Let the model learn the patterns and assign each data point to a specific cluster.

Evaluate the clusters: Assess the quality of the clustering results. There are various evaluation metrics available, such as silhouette score, Dunn index, or within-cluster sum of squares. Choose an appropriate metric based on the nature of the problem and the available ground truth (if any).

Interpret and analyze the clusters: Analyze the characteristics and patterns within each cluster. Examine the centroid or representative points of each cluster and interpret the meaning of the clusters based on domain knowledge. Visualize the clusters using techniques like scatter plots, heatmaps, or parallel coordinate plots.

Iterate and refine: Depending on the evaluation results and domain expertise, iterate on the previous steps to refine the model. This may involve adjusting hyperparameters, feature selection, or exploring different algorithms.

Apply the model: Once you have a satisfactory clustering model, you can apply it to new or unseen data. Assign new data points to the appropriate clusters using the learned model.

The conceptual framework diagram of the proposed model is given below in figure 3.1.

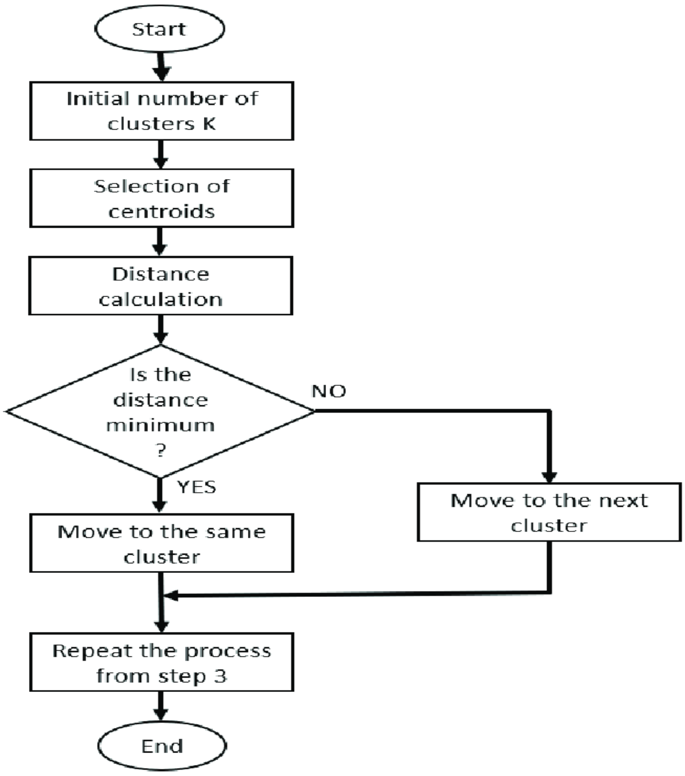


Figure 3.1: Flowchart of proposed model.

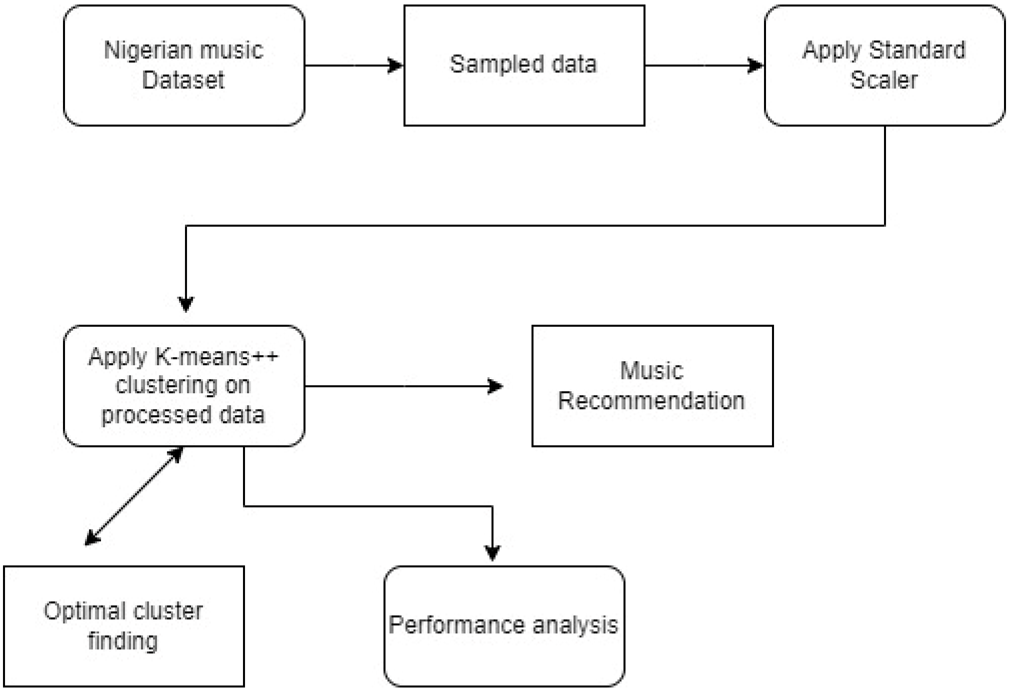


Figure 3.2: Block diagram of the proposed model.

3.2.5 Benefits of proposed model

A music recommendation model offers several benefits for both users and music streaming platforms. Here are some of the key advantages:

1. **Personalized music discovery:** Music recommendation models enable personalized music recommendations based on user preferences, listening history, and behavior. By analyzing user data and employing machine learning algorithms, these models can suggest relevant songs, artists, albums, or playlists tailored to each user's taste. This helps users discover new music that aligns with their preferences and enhances their overall music listening experience.
2. **Increased user engagement:** Music recommendation models can increase user engagement and retention on music platforms. By providing personalized recommendations, users are more likely to spend more time exploring and listening to music, leading to increased user satisfaction and loyalty. When users discover music they enjoy, they are more likely to continue using the platform.
3. **Improved user satisfaction:** Tailored music recommendations improve user satisfaction by reducing the effort required to find music of interest. Instead of manually searching for songs or artists, users can rely on the recommendations to discover new music or revisit favorites. This personalized approach saves time and provides a more enjoyable and hassle-free experience for users.
4. **Discovery of niche or lesser-known music:** Music recommendation models have the potential to expose users to niche or lesser-known artists and genres that they may not have discovered otherwise. By considering not only popular mainstream music but also incorporating users' unique preferences and exploring a broader music catalog, these models can broaden users' musical horizons and facilitate the discovery of diverse music.

5. **Increased user interaction and data collection**: Recommendation models encourage user interaction and engagement with the platform. When users provide feedback on recommended songs (such as thumbs up/down or skipping), it helps the model learn more about their preferences and refine future recommendations. Additionally, user interactions and listening data can be valuable for music platforms to gather insights about user behavior and preferences, aiding in refining their music catalog and enhancing their services.
6. **Promotion of new releases and featured content:** Music recommendation models can play a crucial role in promoting new releases, featured content, or curated playlists. By analyzing user data and behavior, platforms can recommend newly released songs or albums to users who are likely to be interested. This benefits both users who stay updated with the latest music and artists who gain exposure to a targeted audience.

3.3 Dataset Description

A dataset is a collection of related, discrete items of related data that may be accessed individually, in combination, or managed as a whole entity.

The music dataset will be gotten from Kaggle which is the biggest library for data for various fields.

3.4 Data Pre-processing

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. Data preprocessing is a data mining technique that involves transforming raw data into an understandable format.

Our data would be preprocessed through the following steps.

1. **Data Cleaning**: This step involves handling missing values, outliers, and noisy data. Missing values can be imputed or removed, outliers can be detected and treated, and noisy data can be filtered or smoothed depending on the specific characteristics of the dataset.
2. **Data Transformation**: Data transformation aims to normalize the data or change its scale to make it suitable for modelling. Common techniques include feature scaling (e.g., standardization or normalization), logarithmic transformation, or power transformation to reduce skewness.
3. **Feature Selection:** If the dataset contains a large number of features, it may be necessary to select a subset of the most relevant features to reduce dimensionality and improve model performance. Feature selection techniques include embedded method, filter method and wrapper methods.
4. **Feature Encoding:** Categorical features may need to be encoded into numerical representations for modelling. Common techniques include one-hot encoding, label encoding, or ordinal encoding, depending on the nature of the categorical data and the requirements of the model.
5. The elbow method is a technique used in clustering analysis to determine the optimal number of clusters in a dataset. It involves plotting the within-cluster sum of squares (WCSS) against the number of clusters and looking for the "elbow" point, where the rate of decrease in WCSS slows down significantly.

3.5 Performance metrics

There are several performance metrics commonly used to evaluate the quality of clustering results. These metrics provide quantitative measures to assess the effectiveness and accuracy of clustering algorithms. Here are some widely used performance metrics for clustering:

1. **Silhouette score:** The silhouette score measures the cohesion and separation of data points within clusters. It ranges from -1 to 1, where a higher score indicates better clustering results. A positive score indicates that data points are well-matched within their own cluster and poorly matched to neighboring clusters.
2. **Within-cluster sum of squares (WCSS):** WCSS measures the compactness of clusters by summing the squared distances between each data point and its centroid within the cluster. Lower WCSS values indicate better clustering, as it implies that data points are closer to their respective centroids and the clusters are more cohesive.
3. **Davies-Bouldin index (DBI):** The DBI evaluates the separation between clusters by considering both the intra-cluster similarity and the inter-cluster dissimilarity. It computes the average similarity between each cluster and its nearest neighboring cluster, with lower DBI values indicating better clustering.
4. **Calinski-Harabasz index (CHI):** The CHI measures the ratio of between-cluster dispersion to within-cluster dispersion. It evaluates the separation and compactness of clusters, with higher CHI values indicating better clustering. It is computed as the ratio of the sum of between-cluster distances to the sum of within-cluster distances.
5. **Adjusted Rand Index (ARI):** The ARI measures the similarity between the clustering result and an external reference (ground truth) clustering, adjusting for chance agreement. It ranges from -1 to 1, with higher values indicating better agreement between the clustering and the reference clustering.
6. **Normalized Mutual Information (NMI):** NMI quantifies the mutual information between the clustering result and a reference clustering, taking into account the relative cluster sizes. It ranges from 0 to 1, with higher values indicating better agreement between the clustering and the reference clustering.

These metrics can provide insight into different aspects of clustering performance, such as cohesion, separation, compactness, and agreement with ground truth. It's important to choose the appropriate metric based on the specific goals, nature of the data, and available information about the true clustering (if available).

It's worth mentioning that these metrics have their own assumptions and limitations. Therefore, it's advisable to use multiple metrics and consider the characteristics of the dataset and the clustering algorithm being used to obtain a comprehensive evaluation of clustering performance.

CHAPTER FOUR

SYSTEM IMPLEMENTATION AND RESULTS

The Nigerian music recommender system implementation involves several steps. Initially, a dataset containing relevant information is collected. The data is the preprocessed to handle missing values, outliers and normalize the features. Feature selection techniques are employed to identify the most important features for prediction.

Next, a machine learning algorithm is chosen, and the model is trained using the preprocessed data.

4.1 Installation Requirement

4.1.1 Hardware requirement

The minimum hardware requirement for this system is a disk space of at least 4GB. Additional requirements are 1 gigahertz (GHZ) or x86-bit processor, 1 gigabyte (GB) RAM (32-bit), 2 gigabyte (GB) RAM (64-bits), Direct x10 graphics card and a 1024x576 monitor, Windows 7 (32-bit or 64-bit operating system) or Windows 8 (32-bit or 64-bit operating system) or higher versions; Internet explorer of versions 8,9,10 and 11 are suitable. Mozilla Firefox 10.x or a later version are other good options.

For this project, the following system requirement needed to be met:

1. A system with at least Core I3
2. A system equipped with a graphics processing unit (GPU)
3. A RAM of 4GB or greater

4.1.2 Software requirement

Python is the programming language used in this project. Google state that python is a high-level interpreted programming language that was first released in 1991. It is designed to be easy to read and write, with a simple syntax that emphasizes readability and reduces the cost of program maintenance. Python has become a popular language for a wide range of applications, including web development, scientific computing, data analysis, artificial intelligence, machine learning and more.

The following are a few of python programming features:

1. Easy to learn
2. It is an interpreted programming language
3. It provides graphical user interface
4. It is both a portable
5. It is an object-oriented programming language

In this project, libraries and functions from the open-source programming language python were utilized. The following are the open-source libraries utilized:

The software requirements needed for the model include:

i. Jupyter notebook

ii. Anaconda 3

iii. Window operating system/ macOS

iv. Matplotlib

v. Numpy

vi. Seaborn

4.2 General Working of Model

This model's execution was carried out using Jupyter notebook. The model was created in order to carry out music recommendation. The models are presented on the pages as follow:

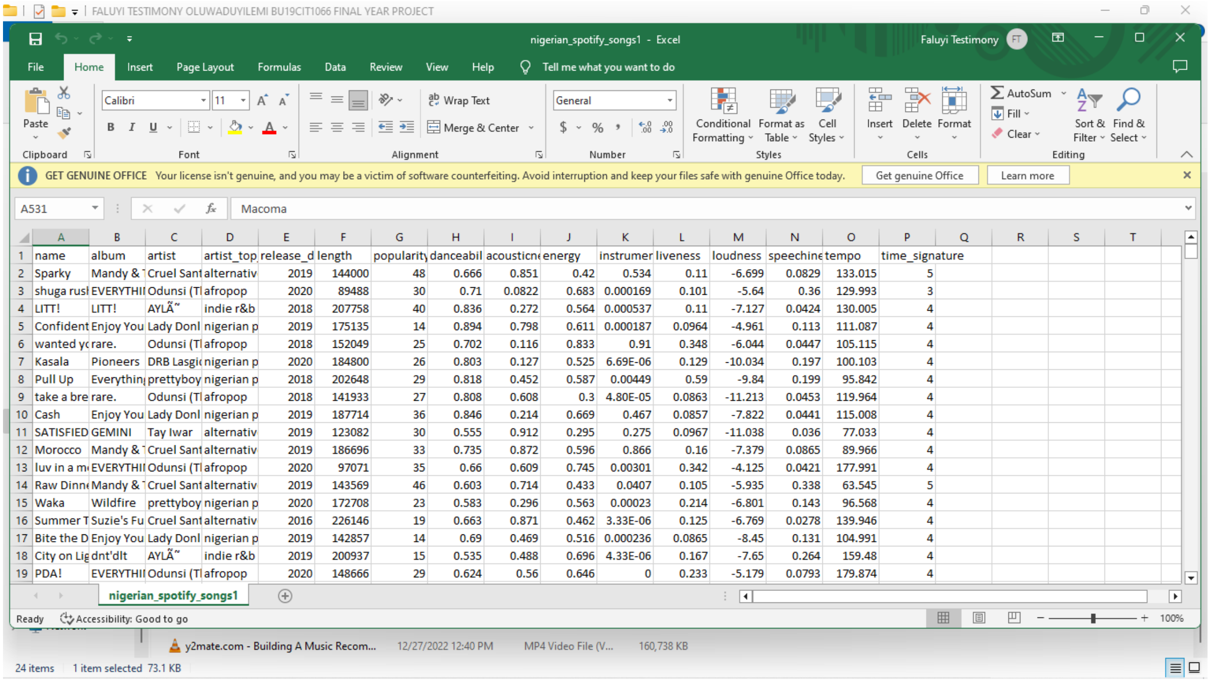
4.2.1 Nigerian music dataset

The dataset for Nigerian music recommendation was downloaded from Kaggle (online data site). The data was stored in a CSV (Comma-Separated Values) format normally used for large datasets due to its flexibility. This way, the data is “raw” and can be processed by different applications.

4.2.2 Loading packages and dataset

Basic python libraries were imported for basic operations and exploratory data analysis. Numpy is a python library used for working with arrays. It has function used in linear algebra and is used to manipulate numbers. Pandas is mainly used for data analysis and manipulation of tabular data in DataFrames. Pandas was used to import the data into Jupyter Notebook in a csv format by writing the following command: “pd. read_csv(‘data’)”. Matplotlib and Seaborn are visualization libraries.

DataFrames and arrays are effectively used by Matplotlib. It treats people as tools and things. It essentially puts out a plot for effective data visualization. Beautiful Python visuals are provided by Seaborn using simple sets of functions. To show output inline, use the '%matplotlib inline' command.



GET GENUINE OFFICE Your license isn't genuine, and you may be a victim of software counterfeiting. Avoid interruption and keep your files safe with genuine Office today.

A531 Macoma

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	name	album	artist	artist_top	release_d	length	popularity	danceabil	acousticne	energy	instrumer	liveness	loudness	speechine	tempo	time_signature
2	Sparky	Mandy & T	Cruel Sant	alternativ	2019	144000	48	0.666	0.851	0.42	0.534	0.11	-6.699	0.0829	133.015	5
3	shuga rusł	EVERYTHII	Odunsi (Tl	afropop	2020	89488	30	0.71	0.0822	0.683	0.000169	0.101	-5.64	0.36	129.993	3
4	LITT!	LITT!	AYLÃ˜	indie r&b	2018	207758	40	0.836	0.272	0.564	0.000537	0.11	-7.127	0.0424	130.005	4
5	Confident	Enjoy You	Lady Donl	nigerian p	2019	175135	14	0.894	0.798	0.611	0.000187	0.0964	-4.961	0.113	111.087	4
6	wanted yc	rare.	Odunsi (Tl	afropop	2018	152049	25	0.702	0.116	0.833	0.91	0.348	-6.044	0.0447	105.115	4
7	Kasala	Pioneers	DRB Lasgi	nigerian p	2020	184800	26	0.803	0.127	0.525	6.69E-06	0.129	-10.034	0.197	100.103	4
8	Pull Up	Everythin	prettyboy	nigerian p	2018	202648	29	0.818	0.452	0.587	0.00449	0.59	-9.84	0.199	95.842	4
9	take a bre	rare.	Odunsi (Tl	afropop	2018	141933	27	0.808	0.608	0.3	4.80E-05	0.0863	-11.213	0.0453	119.964	4
10	Cash	Enjoy You	Lady Donl	nigerian p	2019	187714	36	0.846	0.214	0.669	0.467	0.0857	-7.822	0.0441	115.008	4
11	SATISFIED	GEMINI	Tay Iwar	alternativ	2019	123082	30	0.555	0.912	0.295	0.275	0.0967	-11.038	0.036	77.033	4
12	Morocco	Mandy & T	Cruel Sant	alternativ	2019	186696	33	0.735	0.872	0.596	0.866	0.16	-7.379	0.0865	89.966	4
13	luv in a m	EVERYTHII	Odunsi (Tl	afropop	2020	97071	35	0.66	0.609	0.745	0.00301	0.342	-4.125	0.0421	177.991	4
14	Raw Dinn	Mandy & T	Cruel Sant	alternativ	2019	143569	46	0.603	0.714	0.433	0.0407	0.105	-5.935	0.338	63.545	5
15	Waka	Wildfire	prettyboy	nigerian p	2020	172708	23	0.583	0.296	0.563	0.00023	0.214	-6.801	0.143	96.568	4
16	Summer T	Suzie's Fu	Cruel Sant	alternativ	2016	226146	19	0.663	0.871	0.462	3.33E-06	0.125	-6.769	0.0278	139.946	4
17	Bite the D	Enjoy You	Lady Donl	nigerian p	2019	142857	14	0.69	0.469	0.516	0.000236	0.0865	-8.45	0.131	104.991	4
18	City on Lig	dnt'dlt	AYLÃ˜	indie r&b	2019	200937	15	0.535	0.488	0.696	4.33E-06	0.167	-7.65	0.264	159.48	4
19	PDA!	EVERYTHII	Odunsi (Tl	afropop	2020	148666	29	0.624	0.56	0.646	0	0.233	-5.179	0.0793	179.874	4

y2mate.com - Building A Music Recom... 12/27/2022 12:40 PM MP4 Video File (V... 160,738 KB

Figure 4.1: Nigerian music dataset

4.2.3 Data cleaning and pre-processing

Cleaning and preprocessing data are important processes in preparing a dataset for analysis or machine learning tasks. They entail coping with missing or inconsistent data, outliers, and translating the data into an acceptable format. Here's how data cleaning and preprocessing work:

Data Cleaning: Data cleaning entails identifying and fixing concerns with the dataset's quality and completeness. Its goal is to verify that the data is reliable, consistent, and error-free. Typically, the following tasks are carried out during data cleaning.

Missing Data: Missing data can occur for a variety of reasons, including data input errors or incomplete records. It is critical to identify missing values and devise a strategy for dealing with them. This could include deleting missing data rows or columns, imputing missing values using statistical approaches, or using customized procedures dependent on the type of the data.

Data Preprocessing: Data preprocessing is the transformation of a dataset to make it more appropriate for analysis or machine learning techniques. It seeks to increase data quality while also improving model performance. During data preprocessing, the following tasks are frequently performed:

Data Scaling or Normalization: Data scaling or normalization ensures that all features are on the same scale. When features have distinct ranges or units of measurement, this step is critical. Standardization (z-score normalization) and normalizing to a given range (min-max normalization) are two common scaling strategies.

Feature Encoding: Before being used in machine learning models, categorical variables are often encoded into numerical values. Depending on the nature of the data and the model's requirements, this can be accomplished by one-hot encoding, label encoding, or ordinal encoding.

Feature Transformation: Feature transformation is the process of applying mathematical functions to features in order to generate new representations or derive new features. Transformations that are commonly used include logarithmic, square root, and power transformations, which can aid with skewed distributions or nonlinear relationships.

4.2.4 Exploratory data analysis

Exploratory data analysis, often known as data visualization is a method of evaluating and summarizing data in order to gain insights and understanding of its properties.

In Figure 4.2, the subplot of the top 10 artist occurrence and top 10 album occurrence in the data is shown. Subplot provides a convenient way to organize and present complex visualizations effectively.

In Figure 4.3, the bar chart of the 10 most popular songs in the data is shown. Bar chart are effective for visually comparing the values or frequencies of different categories.

In Figure 4.4, the pie chart shows the percentage of the artist top genres in the data. Pie charts are effective in visually representing the relative portions of different categories in a dataset.

In Figure 4.5, a sunburst chart for top songs in the data is shown. Sunburst charts are useful for visualizing hierarchical data that has multiple levels of categories and subcategories.

In Figure 4.6, the graph of the artist top genre released per year is shown.

In Figure 4.7, the correlation among numeric datatype is shown. Correlation shows the statistical relationship between two or more variables.

In Figure 4.8, the heatmap of missing values in the data is shown. Heatmap is used to analyze relationships, patterns or variations in the data across different dimensions.

In Figure 4.9, the bar chart of top genres in the artist top genre.

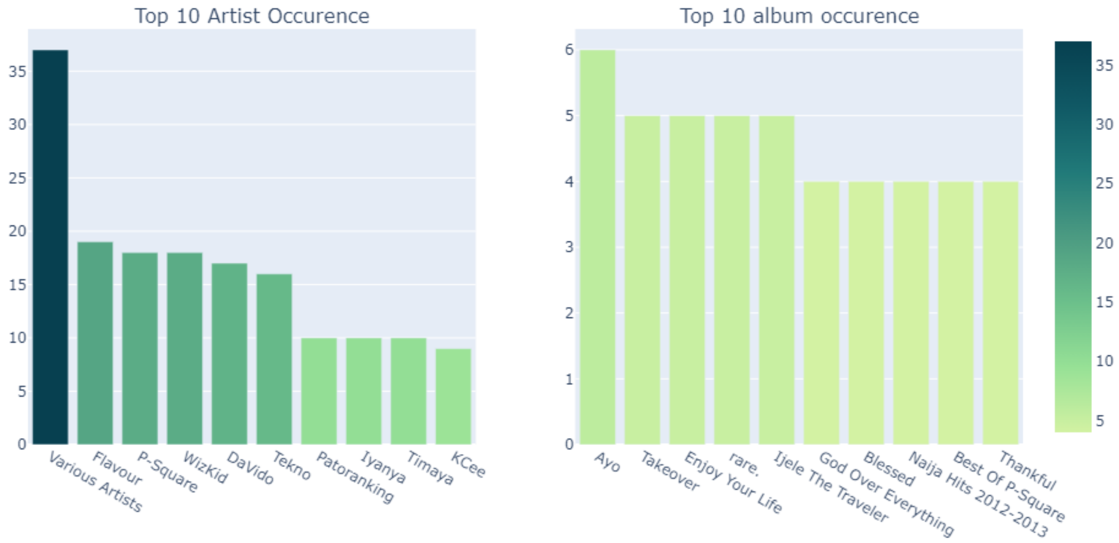


Figure 4.2: Barchart of top 10 artist and album occurrence in data

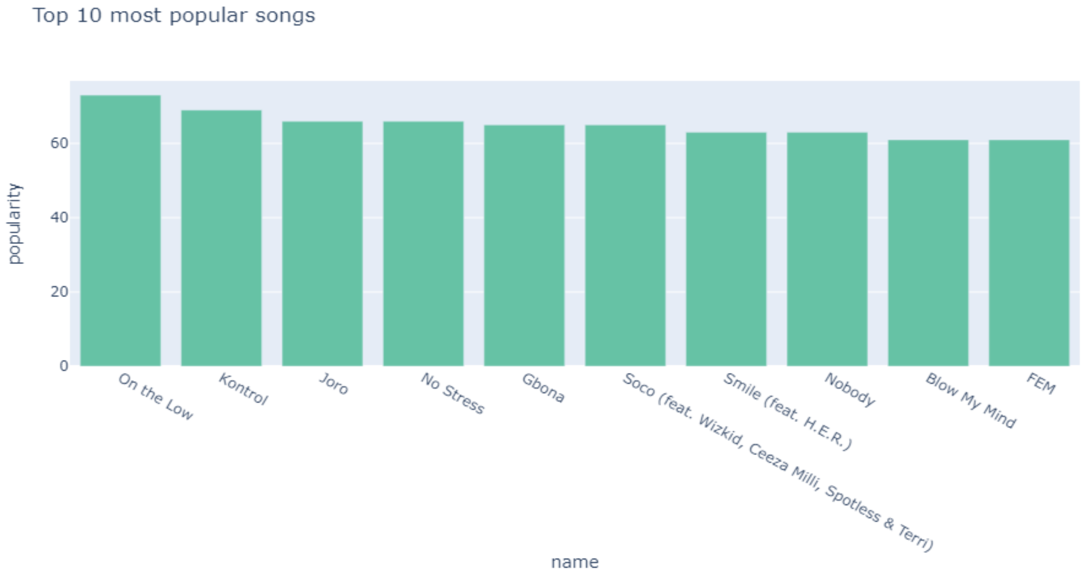


Figure 4.3: Barchart of top 10 most popular songs in data

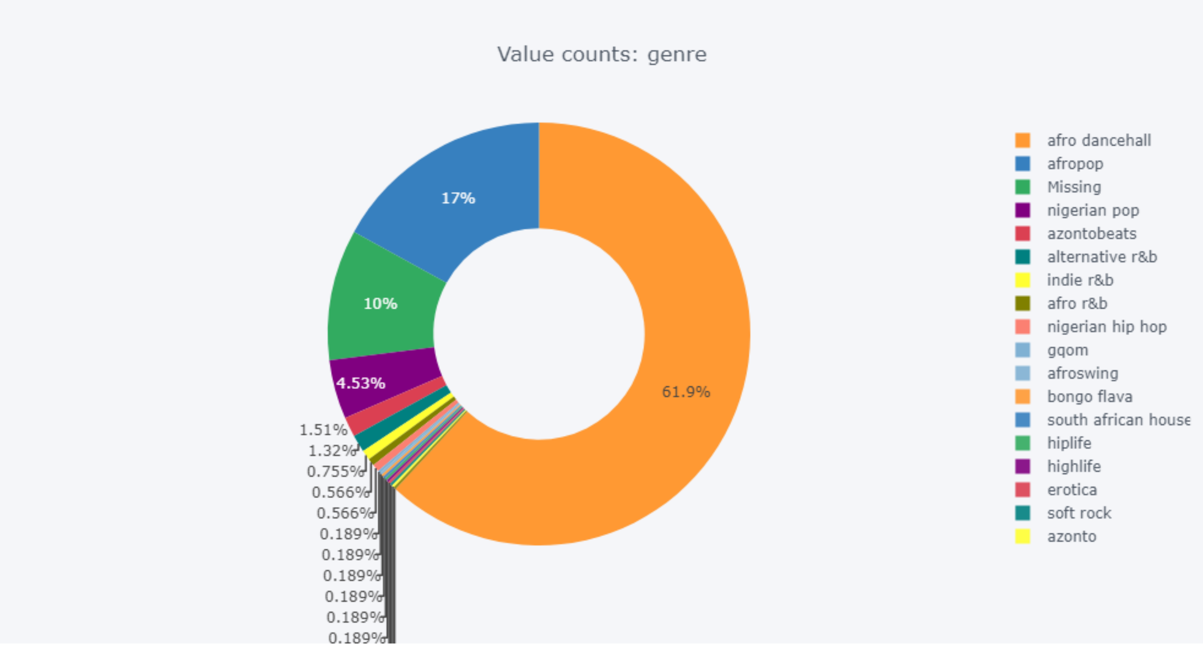


Figure 4.4: Pie chart of artist top genre in data

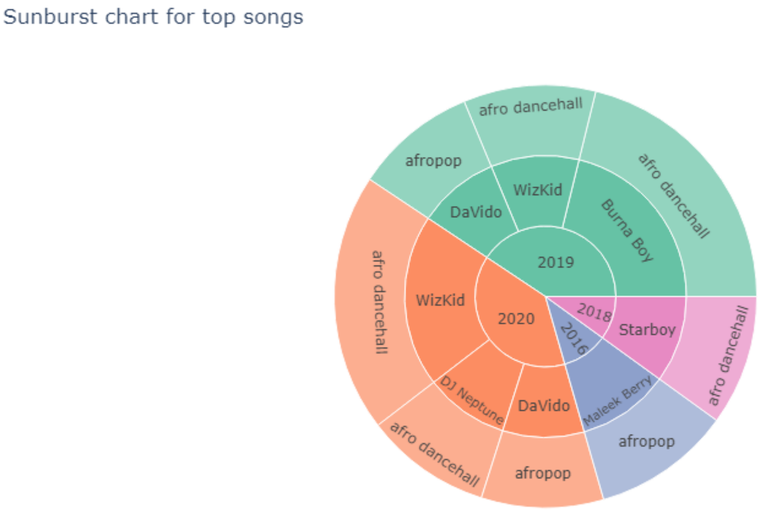


Figure 4.5: Sunburst chart of top songs in data

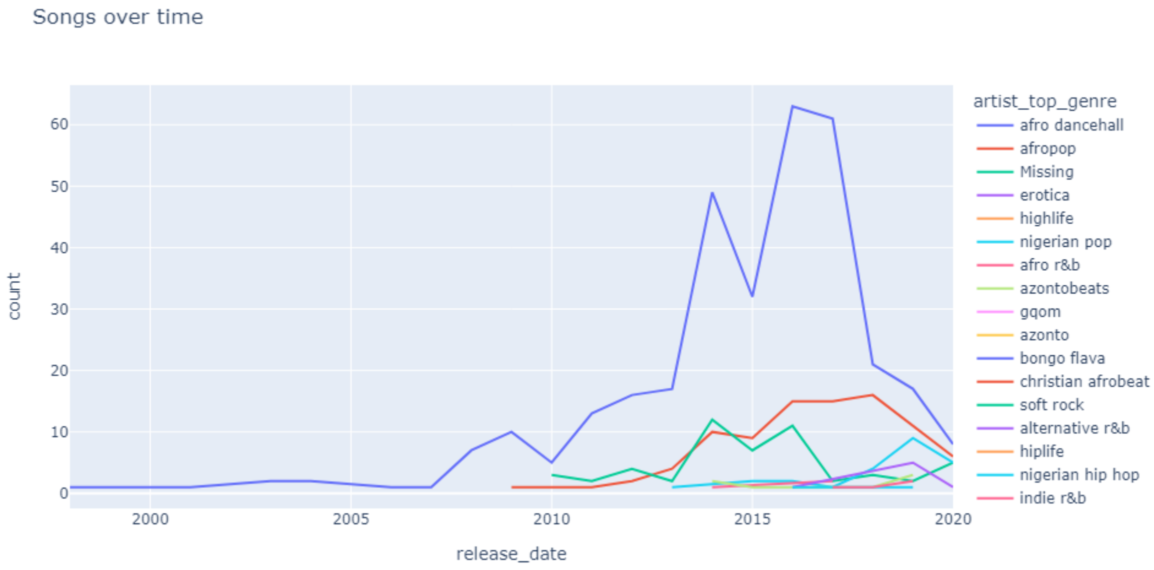
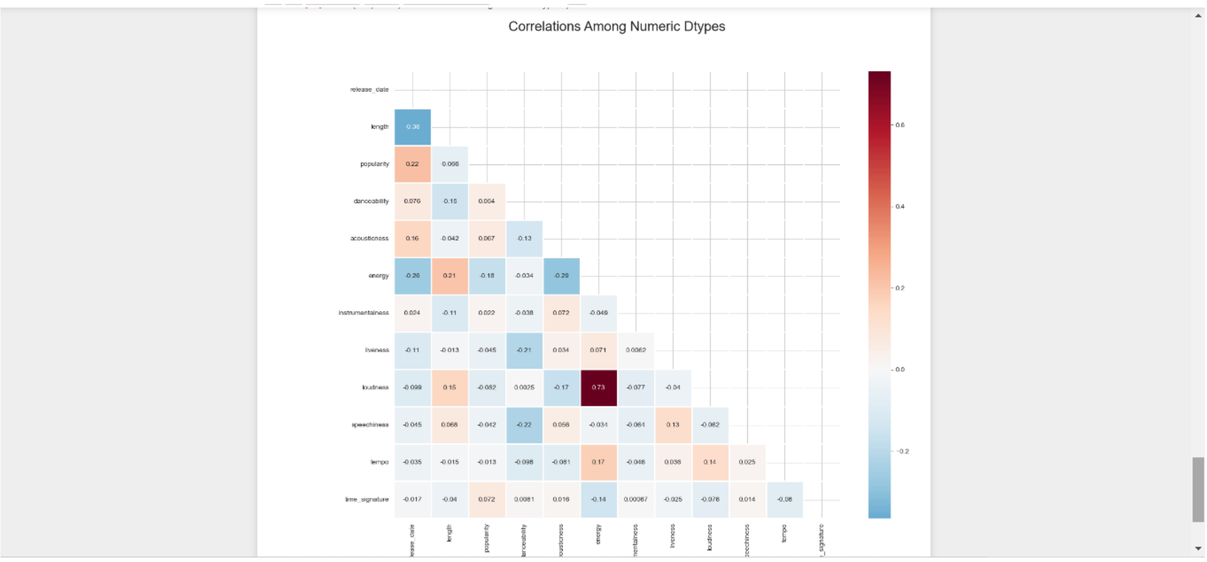


Figure 4.6: Graph of artist top genre release per year.



	lease_date	length	popularity	lanceability	oustioness	energy	nentalness	liveness	loudness	oechiness	tempo	_signature
release_date												
length	-0.36											
popularity	0.22	-0.066										
danceability	0.076	-0.15	0.064									
acousticness	0.16	-0.042	0.067	-0.13								
energy	-0.26	0.21	-0.18	-0.034	-0.29							
instrumentalness	0.024	-0.11	0.022	-0.038	0.072	-0.049						
liveness	-0.11	-0.013	-0.045	-0.21	0.034	0.071	0.0062					
loudness	-0.098	0.15	-0.082	0.0025	-0.17	0.73	-0.077	-0.04				
speechiness	-0.045	0.068	-0.042	-0.22	0.056	-0.034	-0.064	0.13	-0.062			
tempo	-0.035	-0.015	-0.013	-0.098	-0.081	0.17	-0.048	0.038	0.14	0.025		
time_signature	-0.017	-0.04	0.072	0.0081	0.016	-0.14	0.00067	-0.025	-0.078	0.014	-0.08	

Figure 4.7: Correlation among numeric data types in data

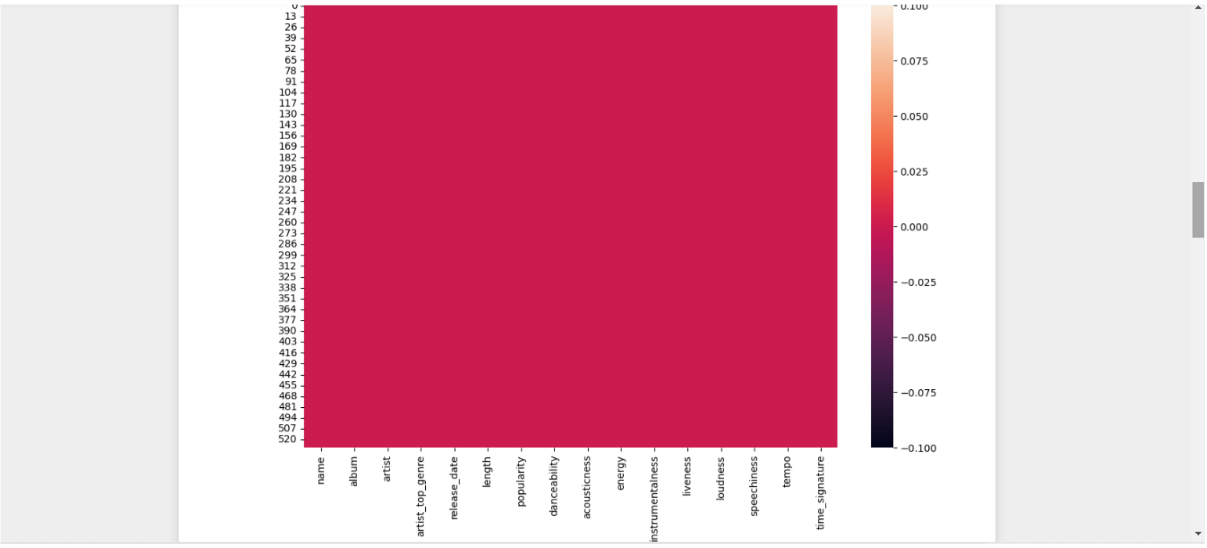


Figure 4.8: Heatmap of missing values in the data

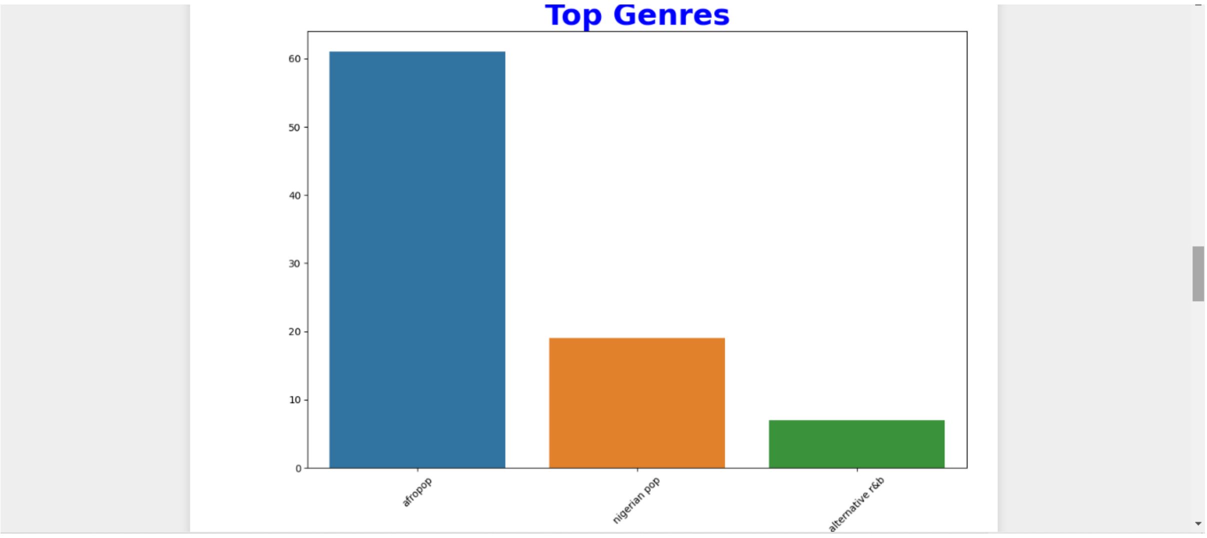


Figure 4.9: Barchart of top genres in data

4.2.5 Machine learning algorithm

KMeans Algorithm was applied on the dataset.

The k-means algorithm is a popular unsupervised machine learning algorithm used for clustering analysis. It aims to partition a given dataset into k distinct clusters based on their similarity. Each cluster is represented by its centroid, which is the mean of all the data points assigned to that cluster. Elbow method was another algorithm that was used to perform the clustering model.

In Figure 4.10, Elbow method used to describe the number of clusters to use is shown. By visually inspecting the plot, you look for the "elbow" or bend in the curve. The elbow point represents the number of clusters where the decrease in WCSS starts to level off. This point is typically considered the optimal number of clusters for the given dataset.

In Figure 4.11, Scatter Plot of clusters after fitting the model is shown, Scatter plot is used to explore patterns, correlations, and trends in the data. In a scatter plot, each data point is represented by a marker, typically a dot, positioned on the x-y plane, where the x-axis represents one variable, and the y-axis represents the other variable.

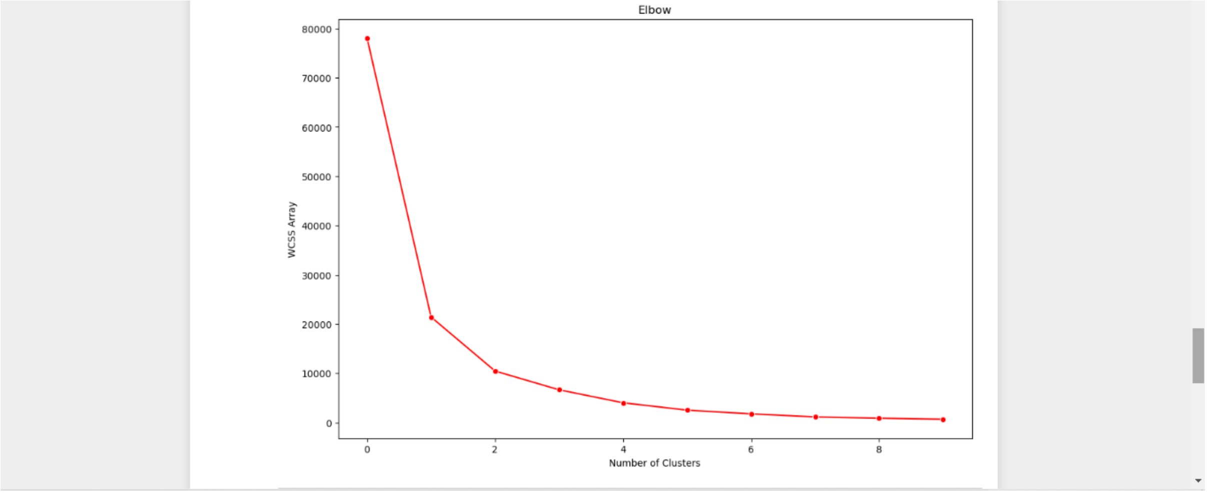


Figure 4.10: Elbow method used to describe the number of clusters to use

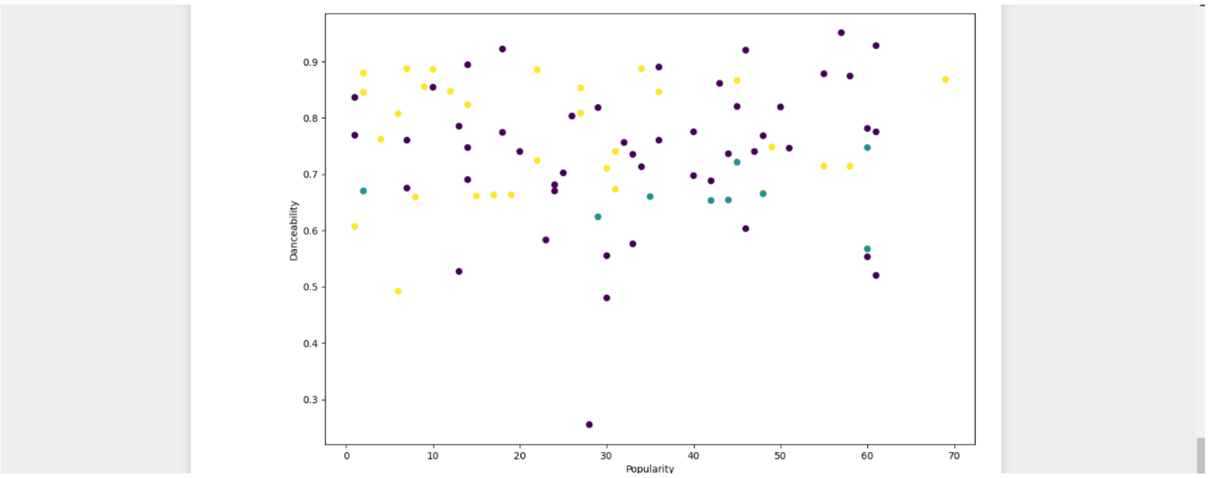


Figure 4.11: Scatter Plot of clusters after fitting the model

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 Summary

The importance of recommending music cannot be overemphasized as it is essential.

As the recommendation model, a K-Means algorithm was chosen. The model was then evaluated using the evaluation metric on the preprocessed data. The model's accuracy measured by the Silhouette score which gave a score of 0.75, which means the model has 75% accuracy.

The project provides recommendations to improve the model's accuracy and efficacy. Suggestions included looking into new features, experimenting with other machine learning methods, optimizing hyperparameters, and using cross-validation techniques.

The Nigerian music recommendation model using clustering and K-means provides personalized music recommendations to users, allowing them to discover and enjoy Nigerian music tailored to their individual preferences. By leveraging clustering techniques, the model can group similar songs or artists together and offer more relevant recommendations, enhancing the overall music listening experience for Nigerian music enthusiasts.

5.2 Conclusion

In conclusion, the Nigerian music recommendation model using clustering and K-means offers a personalized and tailored music discovery experience for Nigerian music enthusiasts. By leveraging user preferences, listening behavior, and clustering algorithms, the model provides relevant recommendations of Nigerian songs, artists, albums.

Through data preprocessing, feature selection, and the application of K-means clustering, the model groups songs or artists into distinct clusters based on their similarities, enabling effective recommendation generation. The resulting recommendations allow users to explore new

Nigerian music aligned with their tastes and preferences, enhancing their music listening experience and satisfaction.

By continuously evaluating and refining the model based on user feedback and relevant evaluation metrics, the model can evolve and improve its recommendations over time. The clustering-based approach ensures that users are exposed to a diverse range of Nigerian music, including both popular and niche or lesser-known artists and genres.

Overall, the Nigerian music recommendation model using clustering and K-means combines data analysis, machine learning techniques, and user-centric design to deliver a personalized and engaging music discovery platform for Nigerian music enthusiasts. It empowers users to explore the vibrant Nigerian music scene and fosters a deeper connection with their favorite artists and genres.

5.3 Recommendation

The project was focused only on developing a Nigerian music recommendation model. It is recommended that:

1. The model is made into a functioning system.
2. There should be more method to use in evaluating the model.

REFERENCES

Bajaj, P., & Suman, U. (2020). A Survey of Music Recommendation Systems and Future Perspectives.

Burke, R., & Ramezani, M. (2001). Matching Recommendation Technologies and Domains. In Proceedings of the 7th International Conference on Intelligent User Interfaces (pp. 367-386).

Cai, L., Zhang, D., & Wang, X. (2011). A Comprehensive Survey of Music Recommendation Systems Based on Content Analysis and Collaborative Filtering.

Chemeque-Rabel, M. (2020). Content-based music recommendation system (Degree Project in Computer Science and Engineering, Second Cycle). Stockholm, Sweden

Chen, L., Choi, J., & Lee, D. (2012). Personalized Music Recommendation: A Survey.

Chen, X., Zhao, H., Wu, D., & Chen, E. (2018). A Survey on Deep Learning in Music Recommendation.

Choi, K., Lim, S., & Lee, S. (2018). Deep Learning for Music Recommendation: Challenges and Opportunities.

Ekstrand, M. D., Kannan, P., & Willemsen, M. C. (2011). Music Recommendation and Discovery in the Long Tail.

Gentile, C., Saggion, H., & Gauch, S. (2011). Exploiting Temporal Dynamics in Music Preferences for Recommendation.

Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative Filtering for Music Recommendation: A Fast Incremental Matrix Factorization Approach.

Ionescu, B., Marques, O., & Muller, M. (2014). Music Recommendation Systems: Challenges and Opportunities.

Jiang, D., Lu, L., Zhang, H., Tao, J., & Cai, L. (2003). Music type classification by spectral contrast feature. https://doi.org/10.1109/icme.2002.1035731

McNee, S. M., Riedl, J., & Konstan, J. A. (2006). Being Accurate is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems. In Proceedings of the 2006 Conference on Human Factors in Computing Systems Extended Abstracts (pp. 1097-1101). New York, NY: Association for Computing Machinery.

Müller, M., & Clausen, M. (2005). Content-Based Music Recommendation Using Audio Signal Analysis.

Nigeria Population (2023) - Worldometer. (n.d.). https://www.worldometers.info/world-population/nigeria-population/

Pérez-Marcos, J., & Batista, V. (2018). Recommender system based on collaborative filtering for Spotify's users. In Proceedings of the 11th International Conference on Theory and Practice of Electronic Governance (pp. 214-220).

Rich, E. (1979). User modeling via stereotypes*. Cognitive Science, 3(4), 329-354.

Sadie, S. (Ed.). (1985). The Cambridge Music Guide. Cambridge University Press.

Serra, X., Gómez, E., & Herrera, P. (2008). Hybrid Music Recommendation by Fusion of Content and Sequential User Behavior.

Schedl, M. (2019). Deep learning in music recommendation systems. Frontiers in Applied Mathematics and Statistics, 5, 44.

Schedl, M. (2016). The lfm-1b dataset for music retrieval and recommendation. In Proceedings of the 2nd International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies (pp. 103-110).

Sturm, B. (2013). The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use. Journal of Intelligent Information Systems, 41(3), 371-406.

Sturm, B. L. (2013). Classification accuracy is not enough. Journal of Intelligent Information Systems, 41(3), 371-406.

Verma, V., Marathe, N., Sanghavi, P., & Nitnaware, P. (2021). Music Recommendation

System Using Machine Learning. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), 7(6), 80-88. doi: 10.32628/CSEIT217615

APPENDIX

A. Clustering python code

```
##import python libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.cluster import KMeans
import warnings
warnings.filterwarnings('ignore')
data=pd.read_csv('nigerian_spotify_songs1.csv')
frame=pd.DataFrame(data)
frame.head()
frame.shape
frame.isnull().sum()
frame.dtypes
frame.columns
frame.columns.unique()
```

```
fig=plt.figure(figsize=(12,8))

sns.heatmap(frame.isnull())

frame=frame[(frame['artist_top_genre']=='afropop')|(frame['artist_top_genre']=='alternative r&b')|

                (frame['artist_top_genre']=='nigerian pop')]

frame=frame[(frame['popularity']>0)]

top_value=frame['artist_top_genre'].value_counts()

fig=plt.figure(figsize = (12, 8))

sns.barplot(x=top_value.index, y=top_value.values)

plt.xticks(rotation=45)

plt.title('Top Genres', color='blue', fontsize=30, fontweight='bold')

lb_en=LabelEncoder()

x=frame.loc[:, ('artist_top_genre','liveness','instrumentalness','speechiness','tempo')]

y=frame['artist_top_genre']

x['artist_top_genre']=lb_en.fit_transform(x['artist_top_genre'])

y=lb_en.transform(y)

km_model=KMeans(n_clusters = 2, random_state = 0)

km_model.fit(x)


# Predict the cluster for each data point..
```

```
y_pred_kmeans=km_model.predict(x)

y_pred_kmeans

from sklearn import metrics

score=metrics.silhouette_score(x, y_pred_kmeans)

score

wcss_array = []



for i in range(1, 11):

    kmeans_model=KMeans(n_clusters=i, init='k-means++', random_state=35)

    kmeans_model.fit(x)

    wcss_array.append(kmeans_model.inertia_)

fig = plt.figure(figsize = (12, 8))

sns.lineplot(wcss_array, marker = 'o', color = 'red')

plt.title('Elbow')

plt.xlabel('Number of Clusters')

plt.ylabel('WCSS Array')

plt.show()

kmeans_model = KMeans(n_clusters = 3)

kmeans_model.fit(x)

km_y_pred = kmeans_model.predict(x)
```

```
fig = plt.figure(figsize = (12, 8))

plt.scatter(frame['popularity'], frame['danceability'], c = km_y_pred)

plt.xlabel('Popularity')

plt.ylabel('Danceability')

plt.show()
```

B. Exploratory Data Analysis python code

```
import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import plotly.graph_objs as go

import plotly

import plotly.express as px

import plotly.figure_factory as ff

from plotly.offline import init_notebook_mode, iplot

from plotly.subplots import make_subplots

import cufflinks

plt.style.use('ggplot')

cufflinks.go_offline()

cufflinks.set_config_file(world_readable=True, theme='pearl')

import warnings

warnings.filterwarnings('ignore')

df = pd.read_csv('nigerian_spotify_songs1.csv')

df.info()
df.isnull().sum()

df.describe()

df.head()

a = df.artist.value_counts().head(10).reset_index()

a.columns = ['artist', 'count']



b = df.album.value_counts().head(10).reset_index()

b.columns = ['album', 'count']


fig = make_subplots(rows=1, cols=2, subplot_titles= ('Top 10 Artist Occurence', 'Top 10 album
occurence'))


fig.add_trace(go.Bar(x= a['artist'], y= a['count'], marker=dict(color=a['count'],
coloraxis="coloraxis")), row=1, col=1)

fig.add_trace(go.Bar(x= b['album'], y= b['count'], marker=dict(color=b['count'],
coloraxis="coloraxis")), row=1, col=2)


fig.update_layout(coloraxis=dict(colorscale='emrld'), showlegend=False)

fig.show()
```

```
c = df.sort_values(by='popularity', ascending=False).head(10)

px.bar(c, x= 'name', y='popularity', color_discrete_sequence=px.colors.qualitative.Set2, title=
'Top 10 most popular songs')

pie = df.artist_top_genre.value_counts()

pie_df = pd.DataFrame({'index':pie.index, 'values': pie.values})

pie_df.iplot(kind='pie', labels= 'index', values= 'values', hole= .5, title="Value counts: genre")

px.sunburst(c, path=['release_date', 'artist', 'artist_top_genre'], values='popularity', title=
'Sunburst chart for top songs',

        color_discrete_sequence=px.colors.qualitative.Set2)

e = df.groupby(['release_date', 'artist_top_genre']).count().reset_index()


px.line(e, x='release_date', y='name', labels= {'name':'count'}, title= 'Songs over time', color=
'artist_top_genre')

cor = df.corr()



mask = np.zeros_like(cor, dtype=np.bool)

mask[np.triu_indices_from(mask)] = True

sns.set_style('whitegrid')

plt.subplots(figsize = (15,12))
```

```
sns.heatmap(cor,
        annot=True,
        mask = mask,
        cmap = 'RdBu_r', ## in order to reverse the bar replace "RdBu" with "RdBu_r"
        linewidths=.9,
        linecolor='white',
        fmt='.2g',
        center = 0,
        square=True)
plt.title("Correlations Among Numeric Dtypes", y = 1.03,fontsize = 20, pad = 40)
```