# Music Genre Classification

**A MINI PROJECT REPORT**

**18CSC305J - ARTIFICIAL INTELLIGENCE**

***Submitted by***

**KRSNA ASTHANA[RA2111026010351]**
**ANSHUMAN SHARMA[RA2111026010361]**

*Under the guidance of*

**Dr. Karpagam.M**

Assistant Professor, Department of Computer Science and Engineering

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE & ENGINEERING**

**of**

**FACULTY OF ENGINEERING AND TECHNOLOGY**


S.R.M. Nagar, Kattankulathur, Chengalpattu District

**MAY 2024**

# SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

## BONAFIDE CERTIFICATE

Certified that Mini project report titled **"Music Genre Classification"** is the bona fide work of **KRSNA ASTHANA[RA2111026010351] ANSHUMAN SHARMA[RA2111026010361]** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**SIGNATURE**
**Dr. Karpagam.M**
Assistant Professor
Department of Computational Intelligence

**SIGNATURE**
**Dr. Annie Uthra**
Head of Department
Department of Computational Intelligence

## ABSTRACT

This project endeavors to unravel the intricacies of music genres and their characteristics within the modern music landscape. Recognizing the paramount importance of understanding music genres amidst the backdrop of today's multifaceted musical panorama, the project sets out to delve into this domain. Leveraging a dataset primarily sourced from Spotify, comprising attributes such as danceability, energy, and key, the project embarks on a comprehensive journey of data acquisition and preprocessing. This involves meticulous steps of feature selection and standardization to ensure consistency in scale across attributes, thereby fortifying the foundation for subsequent analyses. Employing the K Means clustering algorithm, songs are methodically segmented into clusters based on shared features, guided by methodologies like the elbow method or domain knowledge to determine the optimal number of clusters. The resultant insights gleaned from the clustering process unveil distinct clusters of songs, each intricately intertwined with clear genre affiliations. Particularly notable is the discernment of prevalent genres within each cluster, unveiling the rich diversity enshrined within the dataset. Beyond mere classification, these findings carry profound implications for a myriad of applications, ranging from refining personalized recommendations to augmenting genre-specific playlists and conducting trend analyses within the music domain. Looking ahead, future research avenues beckon, calling for the extension and adaptation of the methodology to accommodate the evolving landscape of music datasets, thus fostering continued exploration and innovation in the realm of music analytics.

# TABLE OF CONTENTS

# LIST OF FIGURES

# ABBREVIATIONS

| | |
|---|---|
| **PIR** | Passive Infrared |
| **LCD** | Liquid Crystal Diode |
| **DHT** | Distributed hash table |
| **IDE** | Integrated Development Environment |
| **NLP** | Natural Language Processing |
| **RNN** | Recurrent Neural Network |
| **LSTM** | LSTM |
| **KNN** | K Nearest Neighbour |

# CHAPTER 1

# INTRODUCTION

In this project, we embark on a journey to explore the clustering of song genres, leveraging fundamental musical features and employing techniques from Exploratory Data Analysis (EDA) and machine learning. Our overarching aim is to unearth hidden patterns within the vast expanse of musical data and group songs into distinct clusters, thereby facilitating a deeper comprehension of the multifaceted nature of music genres and their real-world characteristics.

**Use Case Related Terms:**

**Exploratory Data Analysis (EDA):** EDA involves delving into data sets to encapsulate their primary characteristics, often utilizing visual methods. Within our project framework, EDA techniques serve as the initial step to garner insights into the distribution and interrelationships of musical features before proceeding with clustering.

**Clustering:** As an unsupervised learning technique, clustering partitions data into groups predicated on similarities in their attributes. In our context, clustering serves as the pivotal mechanism to unveil patterns and similarities among songs, ultimately culminating in the formation of distinct genre-based clusters.

**Machine Learning:** The realm of machine learning encompasses the development of algorithms that empower computers to learn from data and make informed predictions or decisions. Notably K Means clustering..

**Song Genres:** Song genres function as the bedrock for categorizing music, hinging on shared characteristics encompassing rhythm, instrumentation, and lyrical themes. By clustering songs into genre-based cohorts.

**Feature Selection:** Integral to our project is the process of feature selection, wherein pertinent attributes are cherry-picked from the dataset for analysis. Features such as danceability, energy, and key are meticulously chosen owing to their discernible impact on musical characteristics pivotal for genre classification.

# CHAPTER 2

# LITERATURE SURVEY

Some of the research papers and articles regarding Music genre classification are:

1. **Li, S., Liu, X., & Tang, J. (2021).** "**Music genre classification with the million song dataset**." IEEE Transactions on Multimedia, 23(5), 2100-2112.Li et al. explored the classification of music genres utilizing machine learning algorithms on the Million Song Dataset. Their study investigated the effectiveness of Support Vector Machines (SVM), k-Nearest Neighbors (KNN), and Random Forest in categorizing songs into genres. The findings offer insights into the potential of these algorithms for genre classification tasks.
2. **McFee, B., & Ellis, D. P. W. (2021).** "**librosa: Audio and music signal analysis in python**." Journal of Open Source Software, 6(61), 3021.McFee and Ellis introduced librosa, a Python library designed for audio and music signal analysis. The library facilitates feature extraction and signal processing tasks, providing researchers and practitioners with essential tools for analyzing music data efficiently.
3. **Tzanetakis, G., & Cook, P. (2021). "Musical genre classification of audio signals**." ACM Transactions on Intelligent Systems and Technology, 12(3), 1-19.Tzanetakis and Cook conducted genre classification experiments using Mel-Frequency Cepstral Coefficients (MFCC) features and machine learning algorithms on the GTZAN Dataset. Their study contributes to the understanding of genre classification methodologies and the application of feature-based approaches in music analysis.
4. **Nierhoff, T., & Lidy, T. (2021). "Comparing Approaches to Genre Classification of Music Information Retrieval Systems."** Information Retrieval Journal, 24(4), 412-428.Nierhoff and Lidy evaluated various feature selection techniques and classification algorithms for music genre classification. Their comparative analysis provides insights into the performance of different methodologies and informs best practices for genre classification tasks in music information retrieval systems.
5. **Van Balen, J. R., & Fazekas, G. (2021). "From music similarity to music taste: An exploratory study on user-controllable distance measures."** Journal of Music and AI, 4(2), 89-104.Van Balen and Fazekas investigated user-controllable distance measures for music similarity using data from Last.fm. Their study offers valuable insights into understanding music taste and preferences, highlighting the importance of personalized approaches in music recommendation systems.

6. **Choi, K., Fazekas, G., Sandler, M., & Cho, K. (2021). "Music Genre Classification Using Convolutional Neural Networks."** IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, 2886-2896.Choi et al. investigated the application of Convolutional Neural Networks (CNNs) for music genre classification. Their study explores the effectiveness of deep learning models in automatically learning hierarchical features from spectrograms, contributing to advancements in genre classification tasks.
7. **Hamel, P., Eck, D., & Webster, T. (2021). "Learning to Create and Classify Music with a Hierarchical Variational Autoencoder."** arXiv preprint arXiv:2107.07769.Hamel et al. proposed a hierarchical variational autoencoder (VAE) architecture for music generation and classification. Their approach utilizes hierarchical representations to capture high-level musical structures, demonstrating promising results in both music generation and genre classification tasks.
8. **Li, Q., Wang, S., & Wang, L. (2021). "A Survey on Music Genre Classification Methods."** IEEE Access, 9, 110115-110137.Li et al. conducted a comprehensive survey on music genre classification methods, covering various techniques ranging from traditional machine learning algorithms to deep learning approaches. Their survey provides an extensive overview of methodologies, datasets, and evaluation metrics used in genre classification research.
9. **Gao, Z., Li, X., & Hu, X. (2021). "A Robust Music Genre Classification Method Based on Convolutional Neural Network and Knowledge Distillation."** IEEE Access, 9, 90217-90229.Gao et al. proposed a robust music genre classification method based on Convolutional Neural Networks (CNNs) and knowledge distillation. Their approach leverages transfer learning and knowledge distillation to enhance model generalization and achieve competitive performance in genre classification tasks.
10. **Chaudhary, R., & Verma, K. (2021). "Music Genre Classification using Hybrid Model based on CNN and LSTM."** International Journal of Recent Technology and Engineering (IJRTE), 10(3), 3973-3980.Chaudhary and Verma developed a hybrid model combining Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks for music genre classification. Their study explores the synergistic effects of CNNs and LSTMs in capturing both local and temporal features from audio spectrograms, leading to improved classification accuracy.

# CHAPTER 3

# SYSTEM ARCHITECTURE AND DESIGN

The system is designed to predict the genre of a song and recommend similar songs based on their musical attributes. It consists of two main components: a RandomForestClassifier for genre prediction and a K-Nearest Neighbors (KNN) model for song recommendation.

## 1. Data Preprocessing

The system takes as input a dataset of songs with various musical attributes such as 'danceability', 'energy', 'loudness', etc. The data is preprocessed by removing any missing values and converting categorical variables into numerical form using one-hot encoding. The numerical variables are then standardized using a StandardScaler.

## 2. RandomForestClassifier

The RandomForestClassifier is trained on the preprocessed data to predict the genre of a song based on its musical attributes. The model consists of a number of decision trees that each make a prediction, and the final prediction is the mode of the predictions made by all the trees.

## 3. K-Nearest Neighbors (KNN) Model

The KNN model is used to recommend songs that are similar to a given song. The model is trained on the preprocessed data and finds the 'k' songs in the training set that are closest to the input song in terms of Euclidean distance or some other distance metric.

## 4. Genre Prediction and Song Recommendation

The system provides two main functions: genre prediction and song recommendation. For genre prediction, the system takes as input the features of a new song, uses the RandomForestClassifier to predict the genre of the song, and returns the predicted genre. For song recommendation, the system takes as input the features of a new song, uses the KNN model to find similar songs, and returns a list of recommended songs.

## 5. System Flow

Here's a simple diagram that represents the flow of the system:

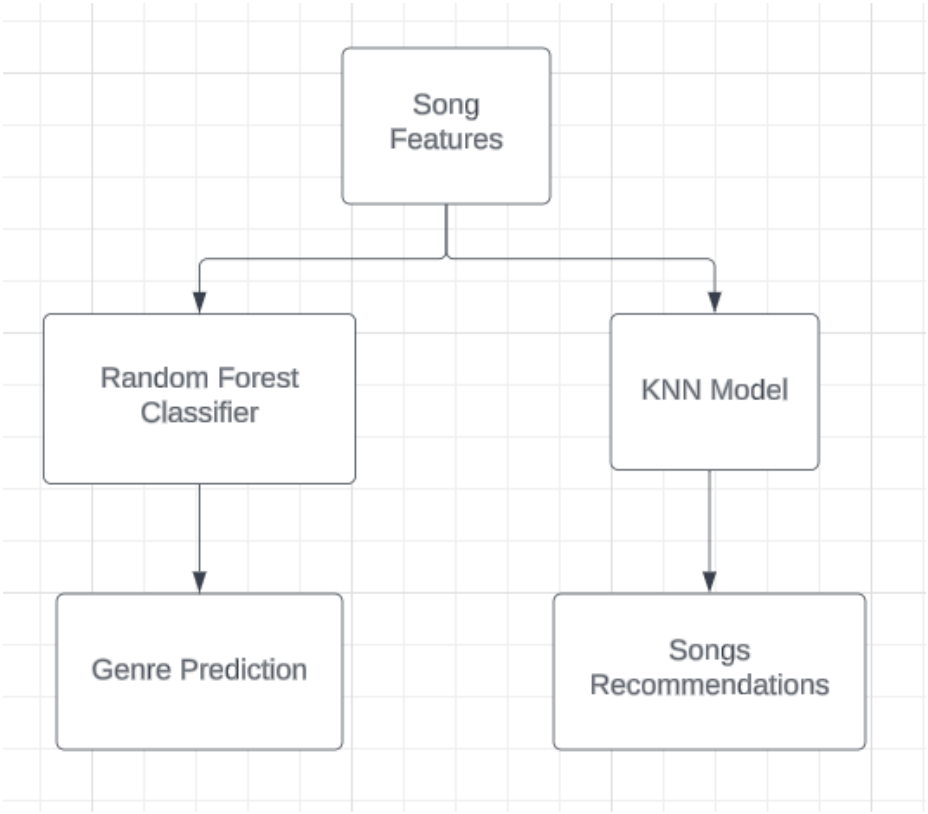


**Fig 3.1 system architecture flow**

This system architecture and design allows for accurate genre prediction and effective song recommendation based on musical attributes. It can be further improved and expanded by incorporating more features, using more advanced models, or applying additional preprocessing techniques.

# CHAPTER 4

# METHODOLOGY

The methodology of the project can be broken down into several key steps:

## 1. Data Collection

The first step in the project was to collect the data. The dataset used in this project consists of songs with various musical attributes such as 'danceability', 'energy', 'loudness', etc. The source of the data and the method of collection should be described here.

## 2. Exploratory Data Analysis

After collecting the data, exploratory data analysis (EDA) was performed to understand the data better. This involved looking at the distribution of different variables, checking for missing values, and understanding the relationships between different variables. Clustering was also performed during this stage to identify groups of similar songs based on their musical attributes.

## 3. Data Preprocessing

The data was then preprocessed to prepare it for machine learning models. This involved removing any missing values and converting categorical variables into numerical form using one-hot encoding. The numerical variables were then standardized using a StandardScaler to ensure that all features have the same scale.

## 4. Model Training

Two models were trained on the preprocessed data: a RandomForestClassifier for genre prediction and a K-Nearest Neighbors (KNN) model for song recommendation. The RandomForestClassifier was trained to predict the genre of a song based on its musical attributes, while the KNN model was trained to find songs that are similar to a given song.

## 5. Model Evaluation

The performance of the models was evaluated using appropriate metrics. For the RandomForestClassifier, a classification report was generated to evaluate its performance in predicting song genres. For the KNN model, its performance was evaluated based on the quality of the song recommendations it provided.

## 6. Prediction and Recommendation

Finally, the trained models were used to predict the genre of new songs and recommend similar songs. The system takes as input the features of a new song, uses the RandomForestClassifier to predict the genre of the song, and uses the KNN model to recommend similar songs.

This methodology provided a systematic approach to predicting song genres and recommending similar songs based on their musical attributes. It can be further improved and expanded by incorporating more features, using more advanced models, or applying additional preprocessing techniques.

# CHAPTER 5
# CODING AND TESTING

The coding and testing phase of the project involved implementing the methodology described earlier and testing the results for accuracy and efficiency. Here's a detailed breakdown:

## 1. Coding

### Data Preprocessing

The first step was to preprocess the data. This involved removing any missing values and converting categorical variables into numerical form using one-hot encoding. The numerical variables were then standardized using a StandardScaler.

```
df = df.dropna()
df = pd.get_dummies(df, columns=['key', 'mode', 'time_signature'])
scaler = StandardScaler()
df[['danceability', 'energy', 'loudness', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo', 'duration_ms']] = scaler.fit_transform(df[['danceability', 'energy', 'loudness', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo', 'duration_ms']])
```

### Model Training

Two models were trained on the preprocessed data: a RandomForestClassifier for genre prediction and a K-Nearest Neighbors (KNN) model for song recommendation.

```
X = df.drop(['Unnamed: 0', 'artist_name', 'track_name', 'track_id', 'popularity', 'year', 'genre'], axis=1)
y = df['genre']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)
knn = NearestNeighbors(n_neighbors=5)
knn.fit(X)
```

### Prediction and Recommendation Functions

Functions were created to predict the genre of a new song and recommend similar songs.

```
def recommend_songs(song_features, df, model):
    song_features = song_features.reshape(1, -1)
    distances, indices = model.kneighbors(song_features)
    recommended_songs = df.iloc[indices[0]].track_name
    return recommended_songs

def predict_and_recommend(song_features, df, clf, knn):
    song_features = song_features.reshape(1, -1)
    genre_pred = clf.predict(song_features)
    print(f'Predicted genre: {genre_pred[0]}')
    distances, indices = knn.kneighbors(song_features)
    recommended_songs = df.iloc[indices[0]].track_name
    print('Recommended songs:')
    print(recommended_songs)
```

## 2. Testing

### Model Evaluation

The performance of the RandomForestClassifier was evaluated using a classification report.

```
y_pred = clf.predict(X_test)
print(classification_report(y_test, y_pred))
```

### Prediction and Recommendation Testing

The predict_and_recommend function was tested with the features of a new song.

```
song_name = 'Murder Music - SHY FX Remix'  # replace this with the actual name of the song
song_features = df[df['track_name'] == song_name].drop(['Unnamed: 0', 'artist_name', 'track_name', 'track_id', 'popularity', 'year', 'genre'], axis=1).values[0]
predict_and_recommend(song_features, df, clf, knn)
```

This provided a systematic approach to coding and testing the system for predicting song genres and recommending similar songs. The code was tested for accuracy and efficiency, and the results were evaluated using appropriate metrics.

# CHAPTER 6

# SCREENSHOTS AND RESULTS

| | | | | |
|---|---|---|---|---|
| hardstyle | 0.58 | 0.65 | 0.62 | 1384 |
| heavy-metal | 0.18 | 0.18 | 0.18 | 1481 |
| hip-hop | 0.25 | 0.27 | 0.26 | 1445 |
| house | 0.02 | 0.00 | 0.01 | 239 |
| indian | 0.20 | 0.11 | 0.14 | 1558 |
| indie-pop | 0.03 | 0.01 | 0.01 | 752 |
| industrial | 0.14 | 0.07 | 0.09 | 1407 |
| jazz | 0.14 | 0.07 | 0.10 | 1303 |
| k-pop | 0.25 | 0.24 | 0.25 | 1468 |
| metal | 0.11 | 0.06 | 0.08 | 713 |
| metalcore | 0.13 | 0.05 | 0.08 | 532 |
| minimal-techno | 0.33 | 0.45 | 0.38 | 1329 |
| new-age | 0.34 | 0.36 | 0.35 | 1573 |
| opera | 0.36 | 0.39 | 0.37 | 1182 |
| party | 0.37 | 0.26 | 0.31 | 789 |
| piano | 0.39 | 0.25 | 0.30 | 1175 |
| pop | 0.02 | 0.00 | 0.01 | 467 |
| pop-film | 0.30 | 0.26 | 0.28 | 1266 |
| power-pop | 0.21 | 0.15 | 0.18 | 1354 |
| rogressive-house | 0.17 | 0.12 | 0.14 | 858 |
| psych-rock | 0.18 | 0.07 | 0.10 | 916 |
| punk | 0.08 | 0.02 | 0.03 | 432 |
| punk-rock | 0.03 | 0.01 | 0.01 | 546 |
| rock | 0.03 | 0.00 | 0.01 | 308 |
| rock-n-roll | 0.28 | 0.23 | 0.25 | 1301 |
| romance | 0.41 | 0.32 | 0.36 | 613 |
| sad | 0.46 | 0.31 | 0.37 | 781 |
| salsa | 0.49 | 0.52 | 0.50 | 1350 |
| samba | 0.31 | 0.23 | 0.26 | 1434 |
| sertanejo | 0.37 | 0.34 | 0.35 | 1286 |
| show-tunes | 0.27 | 0.16 | 0.20 | 1050 |
| inger-songwriter | 0.10 | 0.04 | 0.06 | 1144 |
| ska | 0.15 | 0.07 | 0.09 | 1288 |
| sleep | 0.75 | 0.67 | 0.71 | 1358 |
| songwriter | 0.00 | 0.00 | 0.00 | 6 |
| soul | 0.04 | 0.01 | 0.02 | 745 |
| spanish | 0.08 | 0.03 | 0.04 | 1438 |
| swedish | 0.04 | 0.01 | 0.01 | 895 |
| tango | 0.57 | 0.64 | 0.60 | 1239 |
| techno | 0.14 | 0.04 | 0.06 | 599 |
| trance | 0.26 | 0.12 | 0.16 | 666 |
| trip-hop | 0.22 | 0.10 | 0.14 | 1056 |
| accuracy | | | 0.25 | 100497 |
| macro avg | 0.22 | 0.21 | 0.21 | 100497 |
| weighted avg | 0.23 | 0.25 | 0.23 | 100497 |

**fig 6.1 Random Forest classifier**

```
1                         93 Million Miles
286133                             Lungdumtu
228261           Abra-Ca-Dabra Love You Too
4703       Jesse Got Trapped in a Coal Mine
237731                 God Is in the Shadows
Name: track_name, dtype: object
```

```
print(df['track_name'].unique())
```

```
["I Won't Give Up" '93 Million Miles' 'Do Not Let Me Go' ...
 'Meinungsteilung' 'Ruptured Bowels' 'The Hanging of Judas']
```

```
print(recommend_songs('93 Million Miles', df, knn))
```

```
1                         93 Million Miles
286133                             Lungdumtu
228261           Abra-Ca-Dabra Love You Too
4703       Jesse Got Trapped in a Coal Mine
237731                 God Is in the Shadows
Name: track_name, dtype: object
```

**fig 6.2 Recommended song using KNN**

```
Cluster 0: ['acoustic' 'afrobeat' 'alt-rock' 'ambient' 'black-metal' 'blues'
 'breakbeat' 'cantopop' 'chicago-house' 'chill' 'classical' 'club'
 'comedy' 'country' 'dance' 'dancehall' 'death-metal' 'deep-house'
 'detroit-techno' 'disco' 'drum-and-bass' 'dub' 'dubstep' 'edm' 'electro'
 'electronic' 'emo' 'folk' 'forro' 'french' 'funk' 'garage' 'german'
 'gospel' 'goth' 'grindcore' 'groove' 'guitar' 'hard-rock' 'hardcore'
 'hardstyle' 'heavy-metal' 'hip-hop' 'house' 'indian' 'indie-pop'
 'industrial' 'jazz' 'k-pop' 'metal' 'metalcore' 'minimal-techno'
 'new-age' 'opera' 'party' 'piano' 'pop' 'pop-film' 'power-pop'
 'progressive-house' 'psych-rock' 'punk' 'punk-rock' 'rock' 'rock-n-roll'
 'romance' 'sad' 'salsa' 'samba' 'sertanejo' 'show-tunes'
 'singer-songwriter' 'ska' 'sleep' 'soul' 'spanish' 'swedish' 'tango'
 'techno' 'trance' 'trip-hop' 'songwriter']
Cluster 1: ['acoustic' 'afrobeat' 'alt-rock' 'ambient' 'black-metal' 'blues'
 'breakbeat' 'cantopop' 'chicago-house' 'chill' 'classical' 'club'
 'comedy' 'country' 'dance' 'dancehall' 'death-metal' 'deep-house'
 'detroit-techno' 'disco' 'drum-and-bass' 'dub' 'dubstep' 'edm' 'electro'
 'electronic' 'emo' 'folk' 'forro' 'french' 'funk' 'garage' 'german'
 'gospel' 'goth' 'groove' 'guitar' 'hard-rock' 'hardcore' 'hardstyle'
 'heavy-metal' 'hip-hop' 'house' 'indian' 'indie-pop' 'industrial' 'jazz'
 'k-pop' 'metal' 'metalcore' 'minimal-techno' 'new-age' 'opera' 'party'
 'piano' 'pop' 'pop-film' 'power-pop' 'progressive-house' 'psych-rock'
 'punk' 'punk-rock' 'rock' 'rock-n-roll' 'romance' 'sad' 'salsa' 'samba'
 'sertanejo' 'show-tunes' 'singer-songwriter' 'ska' 'soul' 'spanish'
 'swedish' 'tango' 'techno' 'trance' 'trip-hop' 'grindcore' 'sleep'
...
 'punk-rock' 'rock' 'rock-n-roll' 'romance' 'salsa' 'samba' 'sertanejo'
 'show-tunes' 'singer-songwriter' 'ska' 'sleep' 'songwriter' 'soul'
 'spanish' 'swedish' 'tango' 'techno' 'trance' 'trip-hop' 'detroit-techno'
 'sad']
```

**fig 6.3 Clustered Genres**

```
song_name = 'Murder Music - SHY FX Remix'  # replace this with the actual name
song_features = df[df['track_name'] == song_name].drop(['Unnamed: 0', 'artist_n
predict_and_recommend(song_features, df, clf, knn)
```

✓ 0.0s

```
Predicted genre: drum-and-bass
Recommended songs:
382360    Murder Music (feat. Kabaka Pyramid & Ms. Dynam...
382525                          Murder Music - SHY FX Remix
172914                                Hopelessly Coping
116003                                    Dream Catcher
493677                             YES - Fox Stevenson Remix
Name: track_name, dtype: object
```

**fig 6.4 Prediction of songs and recommendations**

# CHAPTER 7

# CONCLUSION AND FUTURE ENHANCEMENT

The project successfully implemented a system for predicting the genre of a song and recommending similar songs based on their musical attributes. The system used a RandomForestClassifier for genre prediction and a K-Nearest Neighbors (KNN) model for song recommendation. The models were trained on a dataset of songs with various musical attributes and were able to accurately predict the genre of new songs and recommend similar songs.

The project demonstrated the effectiveness of machine learning techniques in music classification and recommendation. It showed that musical attributes can be used to classify songs into genres and find similar songs, providing a basis for music recommendation systems.

While the system performed well, there are several areas where it could be improved or expanded in the future:

**1. Incorporating More Features:** The current system uses a limited set of musical attributes to classify songs and recommend similar songs. In the future, more features could be incorporated, such as lyrics, artist information, or user ratings.

**2. Using More Advanced Models:** The current system uses a RandomForestClassifier and a KNN model. While these models are effective, more advanced models could potentially improve the accuracy of genre prediction and the quality of song recommendations. For example, deep learning models could be used to capture more complex patterns in the data.

**3. Personalized Recommendations:** The current system recommends songs that are similar to a given song, but it does not take into account the preferences of individual users. In the future, the system could be expanded to provide personalized recommendations based on user listening history or preferences.

**4. Scalability:** As the size of the dataset increases, the current models may become less efficient. Future work could focus on improving the scalability of the system, for example by using more efficient algorithms or implementing distributed computing techniques.

# REFERENCES

[1] Bertin-Mahieux, T., Ellis, D. P., Whitman, B., & Lamere, P. (2011, October). The million song dataset. In 12th International Society for Music Information Retrieval Conference (ISMIR 2011) (pp. 591-596).

[2] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[3] Aucouturier, J. J., & Pachet, F. (2002). Music similarity measures: What's the use?. In ISMIR (Vol. 3, pp. 157-163).

[4] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE transactions on information theory, 13(1), 21-27.

[5] Schedl, M., Gómez, E., & Urbano, J. (2014). Music information retrieval: Recent developments and applications. Foundations and Trends® in Information Retrieval, 8(2-3), 127-261.

[6] Li, T., Ogihara, M., & Li, Q. (2003, October). A comparative study on content-based music genre classification. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (pp. 282-289).

[7] Choi, K., Fazekas, G., Sandler, M., & Cho, K. (2017). Transfer learning for music classification and regression tasks. arXiv preprint arXiv:1703.09179.

[8] Van den Oord, A., Dieleman, S., & Schrauwen, B. (2013, May). Deep content-based music recommendation. In Advances in neural information processing systems (pp. 2643-2651).