Music recommendation system based on songs

M.Preetham , P.Vishnu , B.Praveen
Department of Computer Science Engineering
Amrita School of Engineering, Bangalore-35
BL.EN.U4CSE20097@bl.students.amrita.edu, BL.EN.U4CSE 20135@bl.students.amrita.edu,
BL.EN.U4CSE 20130@bl.students.amrita.edu

I. ABSTRACT

In the era of digital music, personalized song recommendations have become an essential part of customer satisfaction. The goal of this project is to enhance the listening experience of music by utilizing many classical machine learning models to select songs based on various attributes extracted from a large dataset. Our primary objective is developing and implementing a robust recommendation system that considers a variety of factors, including user preferences, previous listening behaviour, song attributes (such danceability, energy, and liveliness, for example), and social interactions. To do this, we employ standard machine learning methods including content-based, collaborative, and hybrid filtering.
Index terms: K mean clustering, apriori algorithm, Locality similarity hashing algorithm, music recommendation system.

II. LITERATURE SURVEY

It demonstrates that deep learning is used to make music recommendations based on dance moves, opening the door to uses such as automated choreography and improved human-computer interaction. To build comprehensive systems, future research seeks to establish end-to-end models from motion to music and improve feature extraction for deeper correlations between music and dance. Using machine learning to anticipate preferences and emphasizing the significance of subjective emotions like relaxation and enjoyment in generating such preferences, this study examines the influence of gender, personality, and their interactions on the liking for sad music. It introduces a comprehensive music recommendation system leveraging collaborative, content-based, and hybrid approaches, utilizing machine learning to suggest personalized music based on user preferences and previous interactions with an emphasis on efficiency and accuracy.It presents an emotion-based framework for music recommendations that integrates wearable physiological sensors to identify user emotions and improve on current recommendation systems. Experiments verify the accuracy and integration potential of the framework.

III. INTRODUCTION

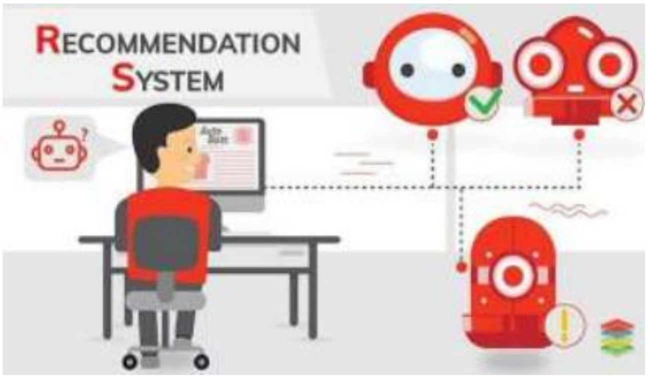

Fig 1. Recommendation System

In modern day virtual music panorama, personalised tune recommendations are pivotal for person pleasure. This venture makes a speciality of enhancing track listening experiences through using numerous conventional gadgets gaining knowledge of models. Our purpose is to create a robust advice machine that elements in consumer alternatives, historical listening styles, music attributes, and social interactions. Leveraging diverse factors which includes collaborative filtering, content material-based totally filtering, and hybrid fashions, we intention to craft a complete gadget that bills for each person-music interactions and inherent track trends.

IV. TECHNIQUES USED

The purpose of this section is to outline the techniques applied during the project. This project involves the following concepts: Apriori algorithm, Locality similarity hashing technique, and Kmean clustering. Anticipatory frameworks, or recommendation frameworks, are essentially frameworks that suggest things to clients or clients to things, and sometimes clients to clients as well. There are various methods for recommending a music to a particular customer.

A.Kmean clustering

Music recommendation systems can benefit from the application of K-means clustering, a popular unsupervised machine learning approach. Within this framework, K-means can identify common patterns within the music data by clustering songs based on similar qualities (e.g., danceability, energy, tempo, etc.). Through a predetermined K cluster segmentation of the dataset, K-means iteratively allocates songs to the closest cluster centroid based on feature similarity. This helps cluster similar songs together in a music recommendation system, which may then be used to generate playlists, organize content, or suggest songs that have similar qualities. By identifying common characteristics among these groups, K-means clustering facilitates the discovery of underlying structures in music data, enabling more specialized and focused song selections.

B. Locality similarity hashing algorithm

The Locality Sensitive Hashing (LSH) technique is a valuable tool for large datasets and is used in recommendation algorithms to approximate song similarity. Similar songs have a high possibility of hashing to the same buckets when they are hashed by LSH. When it comes to song recommendations, LSH will be used to find songs that are comparable based only on their attributes, such as danceability, energy, tempo, and so on. Faster recommendation strategies are made possible by LSH, which hashes songs into buckets and retrieves items from buckets that are like or nearby. This allows for the quicker retrieval of possibly comparable music. By limiting the search space for similar songs, this technique efficiently tackles the difficulty of finding song hints within large music collections, enabling faster and more scalable results.

C.Apriori Algorithm

Apriori is an algorithm specializing in association rule mining, in which we are using in music recommendation systems. Analyses patterns in music data structures to identify common combinations of features (such as rhythm, tempo, energy, etc.) in music A priori works by associative rules that provide relationships between musical characteristics on the oath. In this respect, it helps to identify co-occurring elements in music, showing relationships between musical characteristics. By emphasizing regularities and associative rules, Apriori facilitates musical recommendations based on shared characteristics, in identifying and helping to enforce common combinations suggest similar songs that exhibit comparable patterns or characteristics.

V. About the dataset

An extensive list of the most well-known artists as listed on Spotify is included in this dataset. In comparison to other datasets of a similar nature, this one has a plethora of features. It offers details about the characteristics, level of popularity, and availability of each song on different music outlets. Track names, artist(s) names, duration, kinds, song links, and other audio attributes are all included in the dataset.

V. Data Understanding

Summary of data: It consists of 24 columns and 42306 rows.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42305 entries, 0 to 42304
Data columns (total 22 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   danceability      42305 non-null  float64
 1   energy            42305 non-null  float64
 2   key               42305 non-null  int64
 3   loudness          42305 non-null  float64
 4   mode              42305 non-null  int64
 5   speechiness       42305 non-null  float64
 6   acousticness      42305 non-null  float64
 7   instrumentalness  42305 non-null  float64
 8   liveness          42305 non-null  float64
 9   valence           42305 non-null  float64
 10  tempo             42305 non-null  float64
 11  type              42305 non-null  object
 12  id                42305 non-null  object
 13  uri               42305 non-null  object
 14  track_href        42305 non-null  object
 15  analysis_url      42305 non-null  object
 16  duration_ms       42305 non-null  int64
 17  time_signature    42305 non-null  int64
 18  genre             42305 non-null  object
 19  song_name         21519 non-null  object
 20  Unnamed: 0        20780 non-null  float64
 21  title             20780 non-null  object
dtypes: float64(10), int64(4), object(8)
memory usage: 7.1+ MB
```

Fig 2. It indicates the data types as float, int, object.

The `raw_data.info () usually yields a brief description of the structure of the dataset. The amount of non-null values in each column, column names, data types for each column, and the count of entries are all included. To obtain a fast overview of the data's composition and any potential missing values or data types, this function is frequently used in Python with libraries like Pandas.

```
danceability         0
energy               0
key                  0
loudness             0
mode                 0
speechiness          0
acousticness         0
instrumentalness     0
liveness             0
valence              0
tempo                0
type                 0
id                   0
uri                  0
track_href           0
analysis_url         0
duration_ms          0
time_signature       0
genre                0
song_name        20786
Unnamed: 0       21525
title            21525
dtype: int64
```

Fig 3. It calculates the count of null values per column in the 'raw data'

It determines how many null values are in each dataset column and stores this data in the variable nulls. It also shows the count of null values for each dataset column, highlighting the quantity of missing values in each corresponding column. The output's values each indicate the number of null values in the associated column. When evaluating the quality of the data and choosing how to handle missing information during analysis or preparation, this might be quite important.

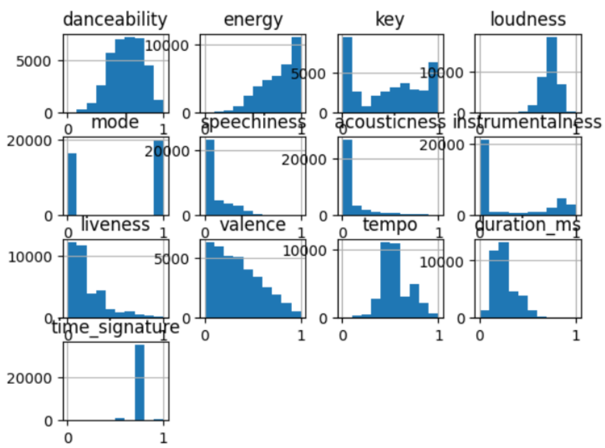

Fig 4. it displays histograms for all numerical columns in 'training data'.

This visualization allows for a quick visual assessment of the data's central tendency, dispersion, and shape of distributions across multiple numerical features, aiding in understanding their characteristics and potential insights for analysis.

VII. Data Visualisation

Correlation matrix: To determine the correlation between every characteristic and determine whether any traits are dependent on any other, a correlation matrix is utilized. The data exhibits a high degree of attribute correlation, as can be shown.

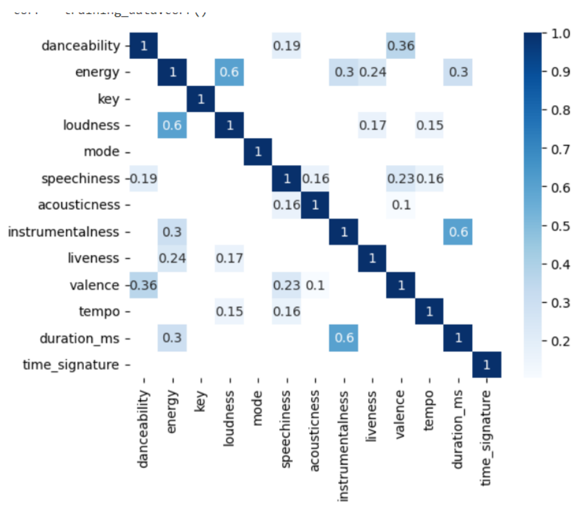

Fig 5. Correlation matrix between the attributes Observation

We see that most correlated variables are Dancebility-Loudness, Energy-Liveness, Mode-Speechiness, Acousticness - Instrumentalness, Valene-Tempo.

A correlation coefficient of 1 indicates a perfect positive correlation that two attributes move in perfect sync. The individual values found in a matrix are represented as colours in a heat map, which is a graphical representation of data. A heat map can be used to depict data patterns and relationships in the context of Recommendation that might not be immediately obvious from looking at the raw data alone.

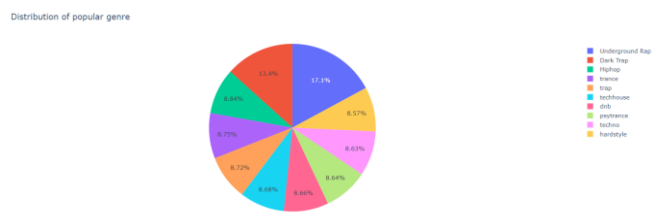

Fig 6. A pie chart of the distribution of popular genres in the raw data dataset.

This pie chart divides into the world of music genres within the "raw data" dataset, revealing fascinating insights into popular trends. The undisputed king of the charts is "Underground Rap" holding a commanding 17.1% share of the data. Hot on its heels come "Dark Trap" "Hip-hop" and "Trance" collectively forming a powerhouse that dominates over half of the entire dataset. While genres like "Techhouse" and "DNB" manage to hold their own with around 8% each, the overall picture paints a clear message: new and emerging styles like "Underground Rap" are taking centre stage. This captivating trend suggests that the "raw data" set might be a treasure trove of music from independent and underground artists, offering a unique glimpse into vibrant and ever-evolving music scene beyond mainstream trends.

VIII. K-means Clustering

K-means clustering is an unsupervised machine learning algorithm used for partitioning a dataset into K distinct, non-overlapping clusters based on similarities in the data points' features. It iteratively assigns points to the nearest cluster centroid and updates centroids until convergence, aiming to minimize the sum of squared distances between data points and their respective centroids. The algorithm requires the number of clusters (K) as input and can be sensitive to initial centroid placement.

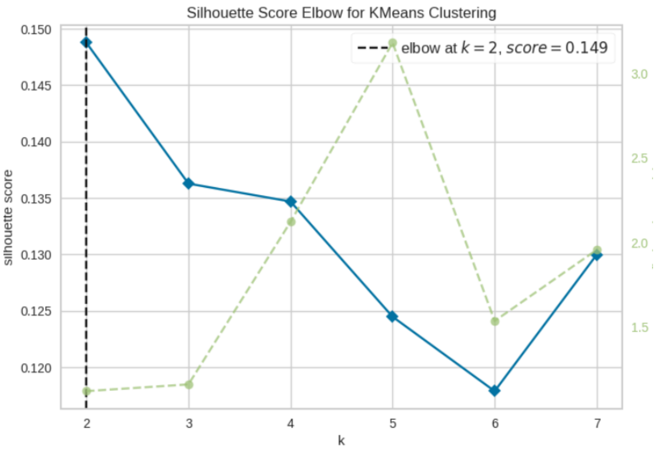

Fig 7. An elbow plot generated for k-means clustering.

Finding the optimal number of clusters (k) is the goal of this. A correlation between the number of clusters and the distortion score—a metric that assesses how well data points are arranged within their designated clusters—is seen in this plot. Ask grows, a dramatic fall in the distortion score usually indicates better grouping. The "elbow" point is eventually marked by the curve's flattening out. The ideal k is at this point, which shows little gain from adding more clusters. The elbow appears around cluster number seven, indicating that the most balanced and successful k-means clustering is probably obtained by dividing your data into seven groups. Recall that the elbow plot can be used in conjunction with other methods such as silhouette analysis or cross-validation to provide a more thorough evaluation of the ideal k.

Clustering and calculating the silhouette score.
Clustering: Clustering is a machine learning technique that forms meaningful subgroups within the facts by grouping record points into clusters based solely on their similarity. Score for Silhouette: The score indicates how well-defined and segregated the clusters are. A higher rating (between -1 and 1) denotes better-described clusters. It measures the nice of grouping. The silhouette score for 'n' clusters is computed by first performing clustering with 'n' clusters using an algorithm such as K-Means, and then calculating the silhouette score for the resulting clustering.

Fig 8. Calculating Silhouette score to the clusters

The outcomes following the computation of the silhouette score for the data in the n clusters, which range from 2 to 6. Three clusters have a high silhouette score, indicating that this is the optimal number of clusters for the data.

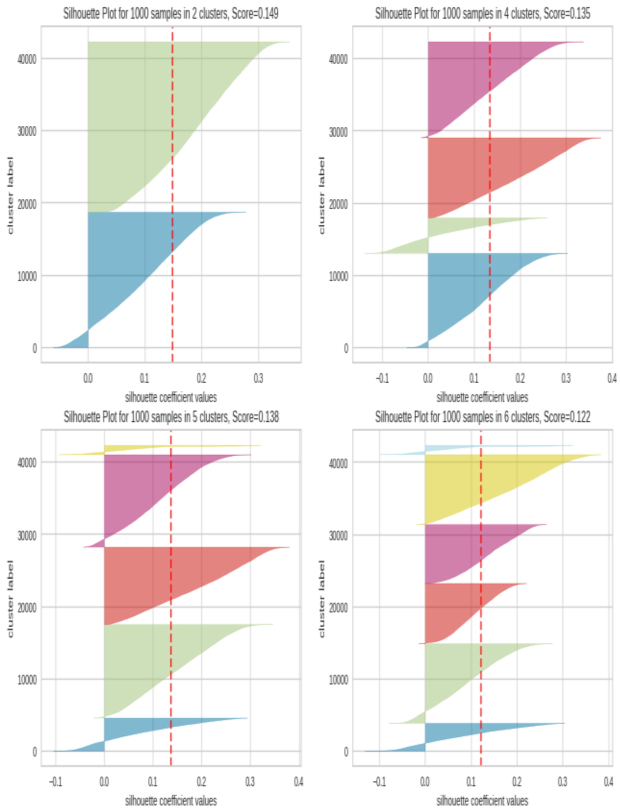

Fig 9. K-Means Clustering with Silhouette Analysis

K-mean clustering's silhouette evaluation reveals that a configuration with four clusters is the most beneficial. This arrangement is backed by a peak average silhouette rating and a closely distributed score around 0, which imply discrete groups within clusters. The choice of four clusters is further supported by a cluster dendrogram, which displays a balanced hierarchical structure and clearly defined separations while reducing at a height of four. All these results highlight how well k-mean clustering with four clusters works, showing that balanced grouping of statistical points and strong clusters work well.

IX. Locality Similarity Hashing Algorithm

By hashing songs to equal buckets with an excessive likelihood, Locality-Sensitive Hashing (LSH) is utilized in recommendation structures to effectively find related music and facilitate rapid similarity searches across large datasets. It's particularly helpful for finding similar objects or files without doing thorough comparisons, such as in approximate closest neighbour searches. By employing comparable device groupings into buckets, assisting tasks like content recommendation, nearest neighbour retrieval, and similarity-based full searches in huge datasets, LSH minimizes computing complexity while accepting a specified degree of error in the results.

```
Song for given features: ['Symbiote', 'Too Good At Goodbyes']
```

Fig10. Nearest song to the given information

By using the song's information—which includes danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentality, liveliness, valence, and tempo—we will be able to locate the closest music. Iteratively encoding each feature's values to a string and updating the MiniHash object is done after first hashing the feature values into minhash objects. A song with specified features from the dataset is queried using the LSH index with a threshold value of o.5 and a permutation value of 128. The result is an output song that matches every feature in the song.

X. Apriori algorithm

A traditional association rule mining method for identifying recurring patterns in data sets is the apriori algorithm. Using a wide first search path to optimize common objects, it finds lists of objects with the lowest support requirements and rediscovers them. The "apriori" attribute eliminates basic, non-computational items that have overhead in discovering important matches. Subsets of common items should also be consistent. readily accepts this technique Market baskets help with decision-making and individualized recommendations based on user actions or concurrent occurrences. They are employed in research and recommendation process types that can uncover interrelationships.

```
46450              (mode)

                                              consequents  antecedent support  \
0                                         (danceability)            1.000000
1                                               (energy)            1.000000
2                                         (danceability)            0.917977
3                                                  (key)            1.000000
4                                                 (mode)            1.000000
...                                                  ...                 ...
46446  (danceability, tempo, energy, key, acousticnes...            0.549462
46447  (danceability, tempo, energy, key, acousticnes...            0.549462
46448  (danceability, tempo, key, acousticness, speec...            0.549462
46449  (danceability, tempo, energy, key, speechiness...            0.549462
46450  (instrumentalness, danceability, tempo, valenc...            0.549462

       consequent support   support  confidence      lift  leverage  \
0                1.000000  1.000000    1.000000  1.000000  0.000000
1                1.000000  1.000000    1.000000  1.000000  0.000000
2                1.000000  0.917977    1.000000  1.000000  0.000000
3                0.917977  0.917977    0.917977  1.000000  0.000000
4                0.549462  0.549462    0.549462  1.000000  0.000000
...                   ...       ...         ...       ...       ...
46446            0.675050  0.353150    0.642719  0.952105 -0.017765
46447            0.675050  0.353150    0.642719  0.952105 -0.017765
46448            0.675050  0.353150    0.642719  0.952105 -0.017765
46449            0.675050  0.353150    0.642719  0.952105 -0.017765
46450            0.675050  0.353150    0.642719  0.952105 -0.017765

       conviction  zhangs_metric
0             inf       0.000000
1             inf       0.000000
2             inf       0.000000
3        1.000000       0.000000
4        1.000000       0.000000
...           ...            ...
46446    0.909507      -0.100439
46447    0.909507      -0.100439
46448    0.909507      -0.100439
46449    0.909507      -0.100439
46450    0.909507      -0.100439
```

Fig 11. Identifying the common patterns

It uses a dataset with certain track functions and the Apriori technique to determine common itemsets and affiliation rules. After binarizing the characteristics of the records, it uses the Apriori technique to find itemsets with a minimum guide of 0.1 as a preprocessing step. It then creates affiliation policies based just on a 0.5 self-belief threshold and prints the discovered rules, which show the connections between musical characteristics.

XI. Recommender System

A recommender system is a data-driven tool that makes predictions and makes suggestions to users about hobbies. It provides personalized hints by interpreting information beyond user behaviour or object characteristics. Recommender systems employ a variety of algorithms to identify user preferences and offer recommendations based on those preferences. These systems locate packages in several areas, such as streaming services, e-commerce, and content recommendations. They enhance user experience by offering personalized options, boosting interaction, and supporting decision-making.

```
Recommended Indices: [42039 39405 41554 40911 40956]
Recommended Songs:
Index 42039:
  Song Name: nan
  Genre: hardstyle

Index 39405:
  Song Name: nan
  Genre: hardstyle

Index 41554:
  Song Name: nan
  Genre: hardstyle

Index 40911:
  Song Name: nan
  Genre: hardstyle

Index 40956:
  Song Name: nan
  Genre: hardstyle
```

Fig 12. Recommended songs.

It uses a Nearest Neighbours model to provide music recommendations based entirely on function targets, including 'genre' and 'song_name'. To locate related songs, it loads a dataset with song properties, extracts relevant features, and fits the Nearest Neighbours model. Using a collection of instance functions, the 'get_recommendations' characteristic finds the songs that are closest in the dataset and outputs their indices, names, and genres. It makes use of the 'Nearest Neighbours' algorithm to calculate similarity, helping to recommend songs that match the given criteria and enhancing the user's experience discovering new music.

XII. Conclusion

To improve track exploration and offer a personalized song recommendation system, the music recommendation system project used a variety of machine learning techniques, including K-means clustering, the Apriori algorithm, and Locality-Similarity Hashing (LSH). K-means clustering allowed for effective user segmentation and categorization by grouping segmented music into discrete groups according to their attributes. To help identify correlations and suggest songs based on shared characteristics, the Apriori algorithm identified common patterns in song properties. LSH made it possible to quickly retrieve similar songs by hashing and indexing music capabilities, which accelerated the advice technology. Together, these approaches improved the machine's capacity to indicate personalized song recommendations, enhancing user narratives and encouraging discovery inside music libraries.

XIII. REFERENCES

[1] W. Gong and Q. Yu, "A Deep Music Recommendation Method Based on Human Motion Analysis," in IEEE Access, vol. 9, pp. 26290-26300, 2021, doi: 10.1109/ACCESS.2021.3057486.

[2] L. Xu, Y. Zheng, D. Xu and L. Xu, "Predicting the Preference for Sad Music: The Role of Gender, Personality, and Audio Features," in IEEE Access, vol. 9, pp. 92952-92963, 2021, doi: 10.1109/ACCESS.2021.3090940.

[3R. Kumar and Rakesh, "Music Recommendation System Using Machine Learning," 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 572-576, doi: 10.1109/ICAC3N56670.2022.10074362.

[4] D. Ayata, Y. Yaslan and M. E. Kamasak, "Emotion Based Music Recommendation System Using Wearable Physiological Sensors," in IEEE Transactions on Consumer Electronics, vol. 64, no. 2, pp. 196-203, May 2018, doi: 10.1109/TCE.2018.2844736.

[5] K. Verbert et al., "Context-Aware Recommender Systems for Learning: A Survey and Future Challenges," in IEEE Transactions on Learning Technologies, vol. 5, no. 4, pp. 318-335, Oct.-Dec. 2012, doi: 10.1109/TLT.2012.11.

[6] E. Aslanian, M. Radmanesh and M. Jalili, "Hybrid Recommender Systems based on Content Feature Relationship," in IEEE Transactions on Industrial Informatics, doi: 10.1109/TII.2016.2631138.

[7] R. Mu, "A Survey of Recommender Systems Based on Deep Learning," in IEEE Access, vol. 6, pp. 69009-69022, 2018, doi: 10.1109/ACCESS.2018.2880197.

[8] K. Yoshii, M. Goto, K. Komatani, T. Ogata and H. G. Okuno, "An Efficient Hybrid Music Recommender System Using an Incrementally Trainable Probabilistic Generative Model," in IEEE Transactions on Audio, Speech, and Language Processing, vol. 16, no. 2, pp. 435-447, Feb. 2008, doi: 10.1109/TASL.2007.911503.

[9] V. Moscato, A. Picariello and G. Sperlí, "An Emotional Recommender System for Music," in IEEE Intelligent Systems, vol. 36, no. 5, pp. 57-68, 1 Sept.-Oct. 2021, doi: 10.1109/MIS.2020.3026000.

[10] F. Fessahaye et al., "T-RECSYS: A Novel Music Recommendation System Using Deep Learning," 2019 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 2019, pp. 1-6, doi: 10.1109/ICCE.2019.8662028.