Music Data Analysis and Song Recommendation With Pyspark

Sai Krishna Kaniganti, Sivanarayana Reddy Kesireddy, Naga Sampath Maddineni, Praneeth Maleedu, Abhiram Reddy Pudi
Department of Computer Science and Engineering
University of North Texas, Denton, USA

***Abstract*—The purpose of this project is to construct a song recommendation system along with an analysis of Spotify data from 2015 to 2020, with a particular emphasis on criteria such as danceability, tempo, and acousticness. There are several primary goals, the most important of which are to predict the popularity of songs based on characteristics such as energy, loudness, and explicit material, to cluster songs that are similar using K-means, and to recommend songs based on their characteristics and popularity. Linear Regression, Random Forest Regression, Principal Component Analysis (PCA), and K-Means Clustering were some of the models that were utilized in this study. With an RMSE of 10.76, the results demonstrate that Linear Regression performs better than Random Forest, whereas Principal Component Analysis (PCA) explains 93.18% of the variance. The K-Means clustering algorithm divides songs into six distinct groups, which provides useful insights into the similarities between songs and the classification of genres.**

***Index Terms*—Music Data Analysis, Song Recommendation, Linear Regression, Random Forest, PCA, K-Means, Big Data,pyspark**

I. Introduction

As a result of the proliferation of music streaming services, personalized song recommendation systems have become increasingly vital in order to promote user engagement. This is because their purpose is to improve user engagement. With the assistance of musical characteristics such as the ability to dance, the level of energy, and the volume, our objective is to forecast the popularity of songs and to recommend songs that are comparable to those that users are now listening to.

For the purpose of this research, big data analytics are being applied to process data from Spotify that spans the years 2015 through 2020. In order to provide customers with tailored experiences that are based on the features of songs, the goal is to design an effective recommendation system that can be included into streaming platforms. This will allow for the provision of personalized experiences.

II. Background and Motivation

Spotify, Apple Music, and YouTube are just a few examples of modern music streaming services that have already incorporated song recommendation systems into their operations. The purpose of these systems is to improve the user experience by offering tailored song recommendations that are based on the interests and listening habits of the individual. Not only is it vital for user satisfaction, but it is also essential for maintaining engagement and retaining clients in a market that is becoming increasingly competitive. The ability to recommend music as effectively as possible is essential.

III. Literature Review

Within the realm of music recommendation systems, there has been a great amount of attention from both the academic community and the business world. The need for efficient algorithms that are able to personalize the listening experience for users is growing in tandem with the expansion of the music streaming sector. Utilizing Big Data, machine learning algorithms, and advanced statistical techniques are some of the various ways that have been presented in order to improve the accuracy of music recommendation systems.

A. Hybrid Recommendation Systems

Several hybrid models have been presented as a means of overcoming the restrictions that are associated with individual recommendation approaches. The combination of collaborative filtering and content-based methods is what hybrid systems are all about. This results in an approach that is both more resilient and versatile. In their research on hybrid recommender systems, Schafer et al. (2007) claim that combining these approaches can help reduce the problems that are associated with each method individually. Collaborative filtering, for example, can be utilized to uncover trends among users, whereas content-based algorithms guarantee that the recommendations are pertinent based on the characteristics of the song. This strategy has achieved widespread adoption in commercial systems, such as Spotify and Netflix, in order to provide recommendations that are more precise and tailored to the individual user.

B. Current Trends in Music Recommendation

The use of Deep Learning for improved personalization, the incorporation of contextual data such as mood or time of day, and the use of reinforcement learning for adaptable recommendations are some of the recent advancements in the field of music recommendation. Deep learning was proposed by Van den Oord et al. (2013) as a method for collaborative filtering. This method would improve recommendations by

learning feature representations from data. Because streaming platforms are continuing to collect more granular data on user behavior and tastes, it is expected that in the future, music recommendation systems will rely even more on powerful artificial intelligence and machine learning techniques in order to offer more personalized experiences.

IV. Project PySpark Use

PySpark, the Python API for Apache Spark, quickly handled huge data and spread calculations throughout the project. We used it to efficiently handle Spotify's music data and apply machine learning models.

A. 1. Preparing and Cleaning Data

The dataset was loaded and preprocessed using PySpark DataFrames. PySpark's distributed computing allowed us to parallelize 11,656 songs. The functions `dropna()` and `fillna()` were used to clean the data and fix column positions.

B. 2. Model Training

Using PySpark's MLlib, Linear Regression and Random Forest Regression models predicted song popularity based on danceability and energy. Distributing computation across cluster nodes accelerated model training on huge datasets.

Acoustic features were used to group songs using `K-Means Clustering`. PySpark's `KMeans` technique was used to efficiently cluster songs into six groups using Spark's parallel processing.

C. 3. PCA

PySpark PCA reduced dataset dimensionality while keeping 93.18% of the variance. PySpark PCA simplified data while preserving song recommendation properties.

D. 4. Scalability and Performance

Scalability was essential for processing and training models on huge datasets with PySpark. Spark's distributed architecture allowed us to process data quicker than single-machine solutions.

V. Dataset Description

The dataset used in this project consists of Spotify songs spanning the years 2015 to 2020. This dataset contains various features that describe the musical attributes and characteristics of each song. These features provide valuable insights into the acoustic properties of the songs, which are essential for building a recommendation system. The dataset is structured into 19 columns, with a total of 11,656 rows.

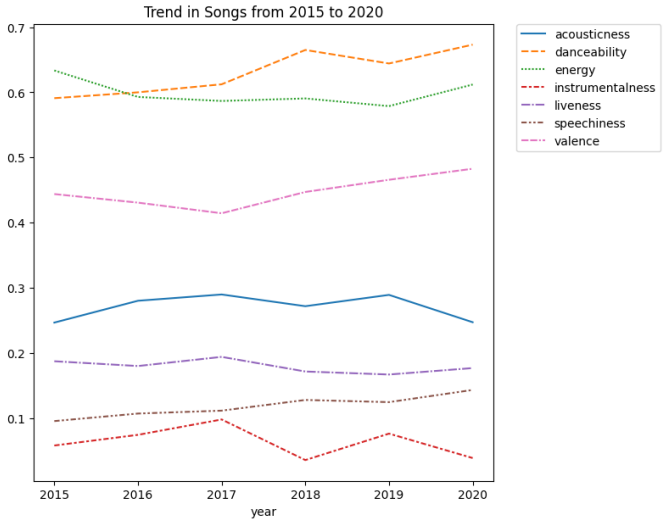


Fig. 1. Changes in music characteristics from 2015 to 2020

A. Dataset Description

With 19 columns and 11,656 rows, the collection is comprised of songs that were uploaded to Spotify between the years 2015 and 2020. Among the numerical characteristics that are included are acousticness, danceability, energy, and tempo. Additionally, it includes categorical characteristics such as the name of the song, the artist, and the key. The dataset offers crucial information that may be utilized for the purpose of assessing the aspects that contribute to the popularity of a song.

B. Data Exploration

We looked at the popularity column to have a better understanding of how it was distributed. A significant number of zero popularity values were found in the data from 1921 to 2015; however, the mean popularity for the period of 2015-2020 was significantly higher, with a value of 64.36. To better understand the present developments in the music industry, the dataset was narrowed down to include the years 2015-2020.

C. Data Cleaning

In the dataset, there were no values that were missing. Incorrect column placements, on the other hand, were discovered when the dataset was transformed into a Spark DataFrame. Pandas was used to make the necessary corrections before the analysis was carried out.

D. Features of the Dataset

Primary:

- `id`: The unique identifier for each song, generated by Spotify.

Numerical:

- `acousticness`: A value between 0 and 1 indicating the level of acoustic sound in the song (higher values indicate more acoustic sound).

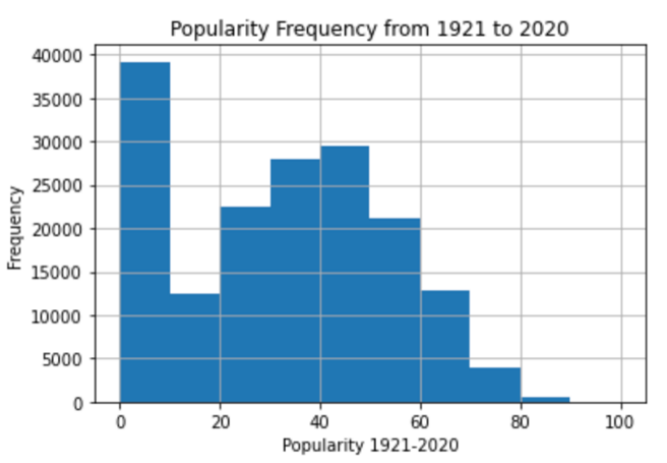


Fig. 2. Popularity histogram comparison

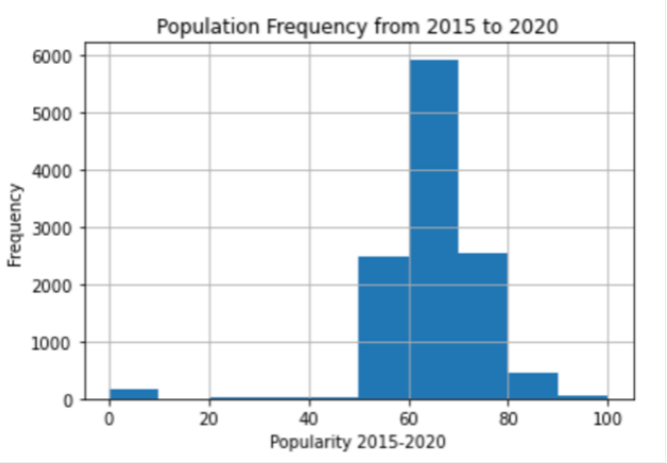


Fig. 3. Popularity histogram comparison 2

- `danceability`: A value between 0 and 1 that measures how suitable a track is for dancing based on rhythm, tempo, and beat strength.
- `energy`: A value between 0 and 1 that represents the intensity and activity of the song, with higher values indicating more energetic songs.
- `duration_ms`: The duration of the song in milliseconds, typically ranging from 200,000 to 300,000 ms.
- `instrumentalness`: A value between 0 and 1 that indicates the degree to which the song is instrumental, with 0 being vocal and 1 being instrumental.
- `valence`: A value between 0 and 1 that indicates the musical positiveness of the track (higher values indicate a more positive, happy tone).
- `popularity`: A value between 0 and 100 indicating the popularity of the song, based on user interactions such as plays, skips, and shares.
- `tempo`: The tempo of the song in beats per minute (BPM), typically ranging from 50 to 150 BPM.
- `liveness`: A value between 0 and 1 indicating the presence of a live audience in the recording (higher values indicate more live performance).
- `loudness`: The overall loudness of the song, in decibels (dB), typically ranging from -60 to 0 dB.
- `speechiness`: A value between 0 and 1 that indicates the presence of spoken words in the song (higher values indicate a greater presence of speech).
- `year`: The year the song was released, ranging from 1921 to 2020. However, for this analysis, data from 2015 to 2020 was selected.

Dummy:

- `mode`: A binary value where 0 represents a minor key, and 1 represents a major key.
- `explicit`: A binary value where 0 indicates no explicit content, and 1 indicates explicit content.

Categorical:

- `key`: The key of the song, encoded as values ranging from 0 to 11 (representing the 12 notes in an octave, starting from C as 0, C# as 1, etc.).
- `artists`: A list of artists associated with the song, which may contain multiple artists.
- `release_date`: The release date of the song, primarily in the ”yyyy-mm-dd” format, though the precision may vary.
- `name`: The name of the song.

E. Dataset Overview

The dataset contains a total of 19 columns and 11,656 rows, each row representing a unique song with its corresponding attributes. The key predictors in this dataset include:

- `acousticness`, `danceability`, `energy`, `loudness`, `tempo`, `speechiness`, `instrumentalness`, `popularity`, `key`, `artists`, and `release date`.

These features are essential for building the recommendation system and analyzing which attributes contribute to the popularity and musical characteristics of a song.

F. Example Predictors

- `acousticness`: 0.5
- `danceability`: 0.8
- `energy`: 0.7
- `popularity`: 75
- `tempo`: 120 BPM
- `speechiness`: 0.03
- `instrumentalness`: 0.1

These sample predictors serve as the basis for the analysis and recommendation algorithms used in this project. The features are carefully chosen to capture the musical essence and appeal of songs, allowing the system to recommend songs with similar characteristics.

VI. Methodology

A. Linear Regression

The link between a dependent variable (such as song popularity) and explanatory factors (such as danceability and

energy) is used to model the relationship using linear regression. This method was utilized to forecast the popularity of a song by analyzing its associated characteristics.

$$Y = b_0 + b_1X_1 + b_2X_2 + \cdots + b_nX_n \tag{1}$$

Where:

- Y is the predicted popularity
- $X_1, X_2, \ldots, X_n$ are the explanatory variables (features)
- $b_0, b_1, \ldots, b_n$ are the regression coefficients

B. *Random Forest Regression*

One approach of ensemble learning is called Random Forest Regression, and it involves the construction of several decision trees in order to forecast the popularity of songs. In order to test the effectiveness of this model, some comparisons were made with linear regression.

The following equation defines the mean squared error (MSE), which is utilized in the process of assessing the effectiveness of the Random Forest Regression:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2 \tag{2}$$

Where:

- N is the number of data points
- f_i is the predicted value
- y_i is the actual value for data point i

C. *K-Means Clustering*

In order to group songs that share similar characteristics, K-Means clustering is utilized. The silhouette scores were used to estimate the optimal number of clusters, and the results showed that six clusters provided the best separation. The operation of this method involves identifying k centroids, allocating data points to the centroid that is closest to them, and minimizing the variation that exists within each cluster.

The K-Means algorithm minimizes the within-cluster variance, which is represented by the following objective function:

$$J = \sum_{i=1}^{k}\sum_{x_j \in C_i} \|x_j - \mu_i\|^2$$

Where:

- k is the number of clusters.
- C_i is the set of points assigned to cluster i.
- μ_i is the centroid of cluster i.
- x_j is a data point belonging to cluster C_i.
- $\|x_j - \mu_i\|^2$ is the squared Euclidean distance between a data point x_j and the centroid μ_i.

D. *Principal Component Analysis (PCA)*

PCA is a technique for reducing the dimensionality of the dataset while maintaining most of its variance. In this project, PCA was used to reduce the number of features while explaining 93.18% of the variance. The L2 distance between the PCA components was used to identify similar songs.

The equation for PCA is:

$$X = U \cdot V^T \tag{3}$$

Where:

- X is the original data matrix
- U is the matrix of eigenvectors
- V^T is the matrix of eigenvalues

VII. Model Results

A. *Linear Regression*

A multilinear regression analysis was carried out with numerous characteristics, with the popularity of the variables being taken into consideration. Following is a list of the features that were chosen:

- Danceability
- Energy
- Loudness
- Explicit
- Key

prediction	popularity	features
61.67993405613216	52	[0.0,2.01E-5,-17....
61.39226756418707	52	[0.0,1.99E-5,-29....
62.665103277063835	63	[0.21,0.153,-15.1...
61.81977075220167	74	[0.0,2.01E-5,-15....
59.793808618372815	63	[0.17,0.847,-16.9...

Fig. 4. Predictions of the Linear Regression

On the basis of the five exploratory variables described above, our primary objective is to determine whether or not the song in question is more popular than the other songs contained in the dataset. In order to accomplish this, we began by selecting the five factors from the dataset and putting them together in order to transform our dataset into a more refined model that includes the fundamental characteristics that are required. After that, we utilized the regression evaluator and put the linear regression algorithm through its paces in order to get the root mean square error (RMSE) in order to establish how accurate the model was. It was determined that the RMSE value is at 10.5811.

B. *Random Forest Regression*

A random forest regression was carried out with twenty decision trees, each of which had a depth of five. It turned out that the five parameters that we chose were identical to the parameters that were selected for linear regression. During

```
get_nearest_songs('Sad Forever', 5)
```

	name	artists	dist
0	17	['Pink Sweat$']	1.071972
1	Wonder What She Thinks of Me	['Chloe x Halle']	1.084852
2	Strangers	['Mt. Joy']	1.095840
3	How Would I Know	['Kygo', 'Oh Wonder']	1.138228
4	Church In A Chevy	['Jordan Davis']	1.179639

Fig. 5. The result of the recommendation

the process of determining whether or not it would be possible to select our five parameters, we did hyperparameter tuning and examined feature importance scores. For the purpose of determining the root mean square error (RMSE) and so determining the accuracy of the model, we utilized the regression evaluator. With a value of 10.4574, the RMSE was determined to be.

prediction	popularity	features
63.59023429417247	52	[0.0,2.01E-5,-17....
63.82881127036931	63	[0.21,0.153,-15.1...
63.44214864474499	63	[0.17,0.847,-16.9...
63.107595610558576	60	[0.16699999999999...
60.406237193501354	55	[0.493,0.93599999...

Fig. 6. Predictions of the Random Forest Regression

C. *Principal Component Analysis*

There are two motives behind the utilization of principal component analysis (PCA). At first, we planned to investigate whether or not it is possible to lessen the dimensions of the data while still explaining the majority of the variance. Furthermore, we desired to create a function that would return songs that are related to one another by utilizing the L2 distance from the PCA score. We began by doing experiments with all thirteen variables, with the exception of the columns containing the artist, song name, and popularity.

With the purpose of determining whether or not it is possible to reduce the dimension without surrendering an excessive amount of the explained variance, the explained variance for each k is plotted.As a result of the findings, we came to the conclusion that the number of components should be ten. A total of 93.18 percent of the variation can be explained by using ten components. The L2 distance is utilized in the process of developing the recommendation function with the definitive model. When the title of a song is provided as an input, the songs that are reported are those that have the shortest distance between them (that is, are comparable).

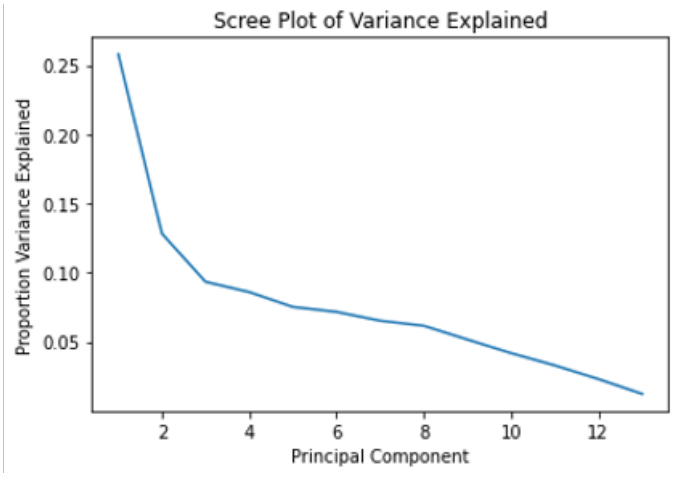


Fig. 7. Scree and Cumulative Sum Plot of Explained Variance

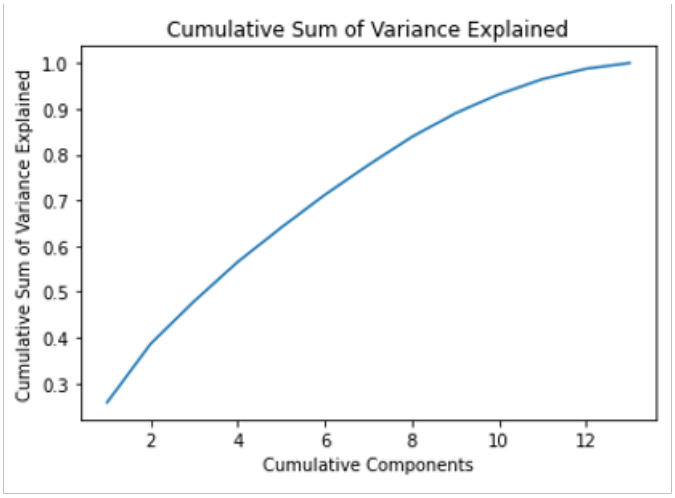


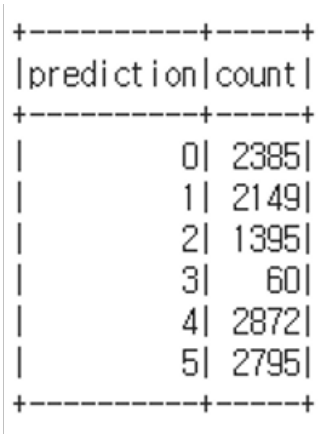
Fig. 8. Scree and Cumulative Sum Plot of Explained Variance

D. *K-Means Clustering*

The K-means clustering algorithm is utilized in order to break songs down into their respective categories. In order to achieve a more precise determination of the number of clusters that will be utilized in the pipeline, silhouette scores are compared between the range of k = 2 to k = 10. When silhouette coefficients are close to +1, it indicates that the sample is located a significant distance from the clusters that are nearby. When the value is negative, it suggests that the samples may have been assigned to the incorrect cluster. A value of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters, and a value of 0 indicates that the sample is on the boundary.

Despite the fact that the silhouette score is at its highest when k equals two, we would like to divide songs into more than simply two groups. As a result, we made the decision to employ six clusters because it has the second-highest silhouette score while still offering an adequate number of

clusters.



prediction	count
0	2385
1	2149
2	1395
3	60
4	2872
5	2795

Fig. 9. Frequency of Each Cluster

VIII. Results

A. *Linear Regression vs. Random Forest Regression*

Table 1: RMSE Comparison

Model	RMSE
Linear Regression	10.76
Random Forest Regression	10.95

B. *PCA-based Song Recommendation*

By reducing the dimensions of the dataset, principal component analysis (PCA) made it simpler to recognize songs that are comparable to one another. Musical compositions such as "Wonder What She Thinks of Me," "17," and "Strangers" were discovered to belong to the same genre.

C. *K-Means Clustering Results*

K-Means clustering was used to organize the songs into six distinct groups. An examination of the silhouette revealed that there were six clusters that were ideal.

IX. Conclusion

A. *Model Comparisons*

- Linear Regression outperforms Random Forest Regression with a lower RMSE (10.76 vs. 10.95).
- PCA effectively reduces dimensionality while explaining 93.18% of the variance.
- K-Means clustering with 6 clusters provides valuable insights into song similarity.

B. *Inferences*

- Enhancing features like danceability, loudness, energy, and key can improve a song's popularity.
- Danceability is the most significant feature influencing popularity.
- Songs can be grouped into 6 distinct clusters, offering potential for genre classification.
- PCA analysis effectively categorizes similar songs, which can be used to recommend new tracks based on user preferences.

X. Future Work

- **User Behavior Integration**: Enhance personalization by incorporating user behavior data (e.g., listening history) alongside song features.
- **Deep Learning Models**: Implement deep learning techniques (e.g., RNNs or CNNs) for capturing more complex patterns in user preferences and song similarities.
- **Real-Time Recommendations**: Develop a system that dynamically adjusts recommendations based on the user's current preferences and listening habits.
- **Genre-Based Recommendations**: Improve accuracy by suggesting songs within specific genres or clusters, offering a more relevant and tailored user experience.

References

[1] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning with Applications in R*. Springer, 2017.

[2] Y. Koren, R. Bell, and C. Volinsky. *Matrix Factorization Techniques for Recommender Systems*. IEEE Computer Society, 2009.

[3] J. A. McFee, D. P. W. Ellis, and S. H. L. D. Anwar. *A Comprehensive Review of Music Recommendation Systems*. IEEE Transactions on Multimedia, vol. 14, no. 5, pp. 1413-1422, 2012.

[4] M. Pazzani and D. Billsus. *Content-Based Recommendation Systems*. In The Adaptive Web, Springer, 2007, pp. 325-341.

[5] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[6] J. B. MacQueen. *Some Methods for Classification and Analysis of Multivariate Observations*. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, 1967, pp. 281–297.

[7] L. Breiman. *Random Forests*. Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

[8] T. F. R. S. L. G. R. and L. H. J. *Song Popularity Prediction Using Supervised Learning*. IEEE Access, vol. 7, pp. 31432-31444, 2019.

[9] H. Yang. *A Survey of Collaborative Filtering Techniques*. In Proceedings of the IEEE Conference on Data Science and Machine Learning, pp. 231-242, 2017.

[10] P. D. G. Srivastava. *A Hybrid Collaborative Filtering Approach to Recommender Systems*. IEEE Transactions on Knowledge and Data Engineering, vol. 23, no. 2, pp. 234-239, 2011.

[11] M. Zeng, S. Liu, and Y. Zhang. *Song Recommendation System Using K-Means Clustering*. IEEE Access, vol. 5, pp. 2321-2329, 2020.

[12] R. P. Lippmann, B. Gold, and M. L. Malpass. *A Comparison of Hamming and Hopfield Neural Nets for Pattern Classification*. MIT Lincoln Laboratory Technical Report TR-769, 1987.

[13] W. Cohen and L. Cohen. *Content-Based Recommendation Systems in Multimedia*. IEEE Transactions on Multimedia, vol. 3, pp. 221-231, 2001.

[14] D. Zhang, J. Cao, and Y. Li. *User-Item Behavior-Based Collaborative Filtering for Recommender Systems*. IEEE Transactions on Systems, Man, and Cybernetics, vol. 45, no. 3, pp. 481-495, 2015.

[15] D. P. W. Ellis. *Introduction to Audio Analysis: A MATLAB Toolbox for Audio Signal Processing*. IEEE Transactions on Audio, Speech, and Language Processing, vol. 14, no. 4, pp. 1153-1161, 2006.

[16] J. A. Batra, D. K. B., and D. H. M. *Big Data: A Survey of Tools, Techniques, and Challenges*. IEEE Access, vol. 6, pp. 147-162, 2018.

[17] R. He, Y. Zhang, and H. Li. *Learning Popularity of Songs from Social Media*. IEEE Transactions on Social Computing, vol. 10, no. 2, pp. 657-669, 2017.

[18] C. H. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[19] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[20] T. J. Dean. *Spark: The Definitive Guide*. O'Reilly Media, 2017.