CSE343: Machine Learning - Project Interim Report
Music Recommendation System Using Machine Learning

Aniket Kanojia
Ashutosh Gera
Piyush Kumar
Tushar Chandra

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Motivation

- Our project aims to create an innovative music recommendation system that stems from our love for music and the desire to continually explore new artists and tones.

- Existing music recommendation systems have limitations, including limited diversity in recommendations, a bias towards popular content, and a struggle to effectively incorporate user feedback. Our goal is to overcome these issues by creating an unbiased system that prioritizes analyzing audio features and user preferences.

- Limitations with existing systems:
  - Limited diversity
  - Biased towards popular content
  - Does not effectively incorporate user feedback.

- By creating a personalized recommendation system, we aim to enhance the music listening experience for everyone and understand how recommendation algorithms work in a real-world context.

Literature review

- The literature review highlights the significance of various studies in shaping a music recommendation system, addressing key aspects such as data processing, feature extraction, user preference models, classification techniques, clustering methods, and real-time adaptation.

- One influential study suggests a dynamic K-means clustering algorithm for personalized music recommendations. This algorithm adapts the number of clusters dynamically and recommends music based on these clusters. Unlike traditional systems, it caters to users who have diverse music genre preferences.The dynamic K-means clustering algorithm aims to recommend music that aligns closely with a user's preferences, even when those preferences span multiple music genres. This approach improves the accuracy of recommendations.

- Another important paper, authored by M. G. Galety and team, focuses on a personalized music recommendation model using machine learning. They explore Count Vectorizer and Cosine similarity techniques to connect music with individual preferences, ultimately offering suggestions based on a user's music history.

- D. Lin and S. Jayarathna, "Automated Playlist Generation from Personal Music Libraries," 2018 IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, UT, USA, 2018, pp. 217-224, doi: 10.1109/IRI.2018.00039. They accomplish the automatic generation of playlists by combining the use of music analysis tools and clustering algorithms from the field of machine learning. They use **K-Means, Affinity Propagation and DBSCAN** algorithms.

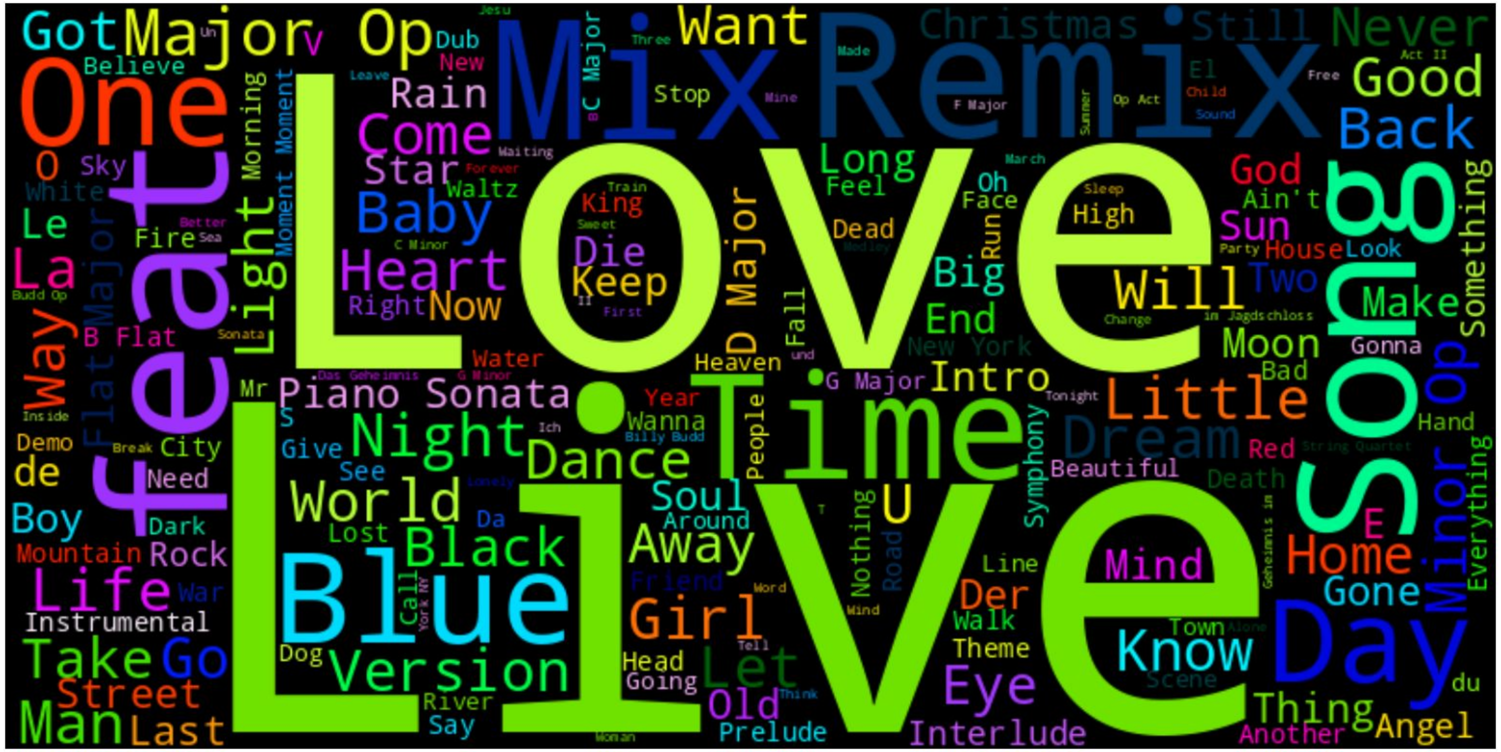
Literature review

Links to Research papers are:

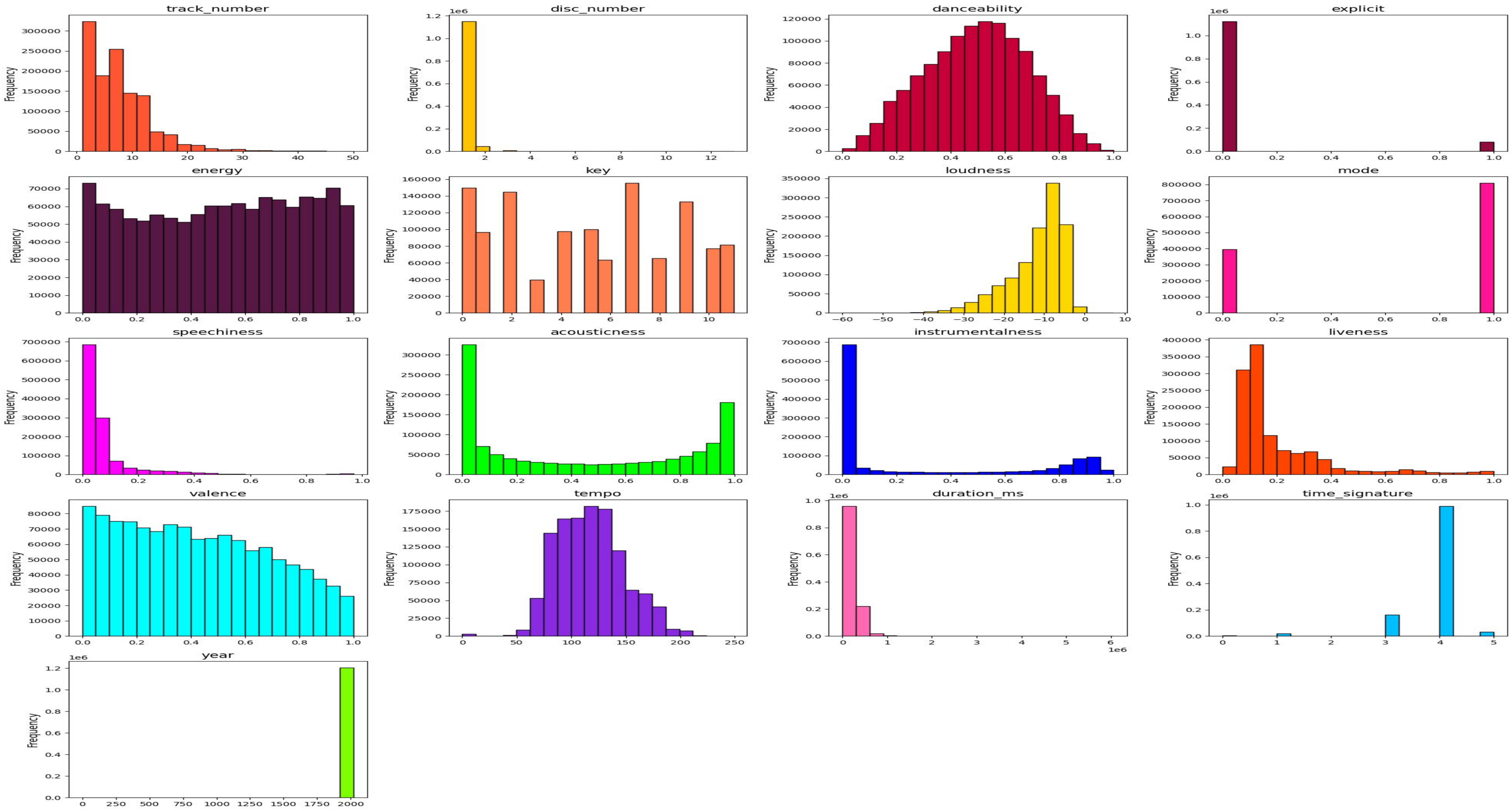
1. A Music Recommendation System with a Dynamic K-means Clustering Algorithm
2. M. G. Galety, R. Thiagarajan, R. Sangeetha, L. K. B. Vignesh, S. Arun and R.
3. Krishnamoorthy, "Personalized Music Recommendation model based on Machine Learning," 2022 8th International Conference on Smart Structures and Systems (ICSSS), Chennai, India, 20
4. D. Lin and S. Jayarathna, "Automated Playlist Generation from Personal Music Libraries," 2018 IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, UT, USA, 2018, pp. 217-224, doi: 10.1109/IRI.2018.00039.

Dataset Description

- An analysis of a music dataset using a word cloud reveals that song titles frequently contain words such as 'Live,' 'Love,' 'feat,' 'song,' 'time,' among others.

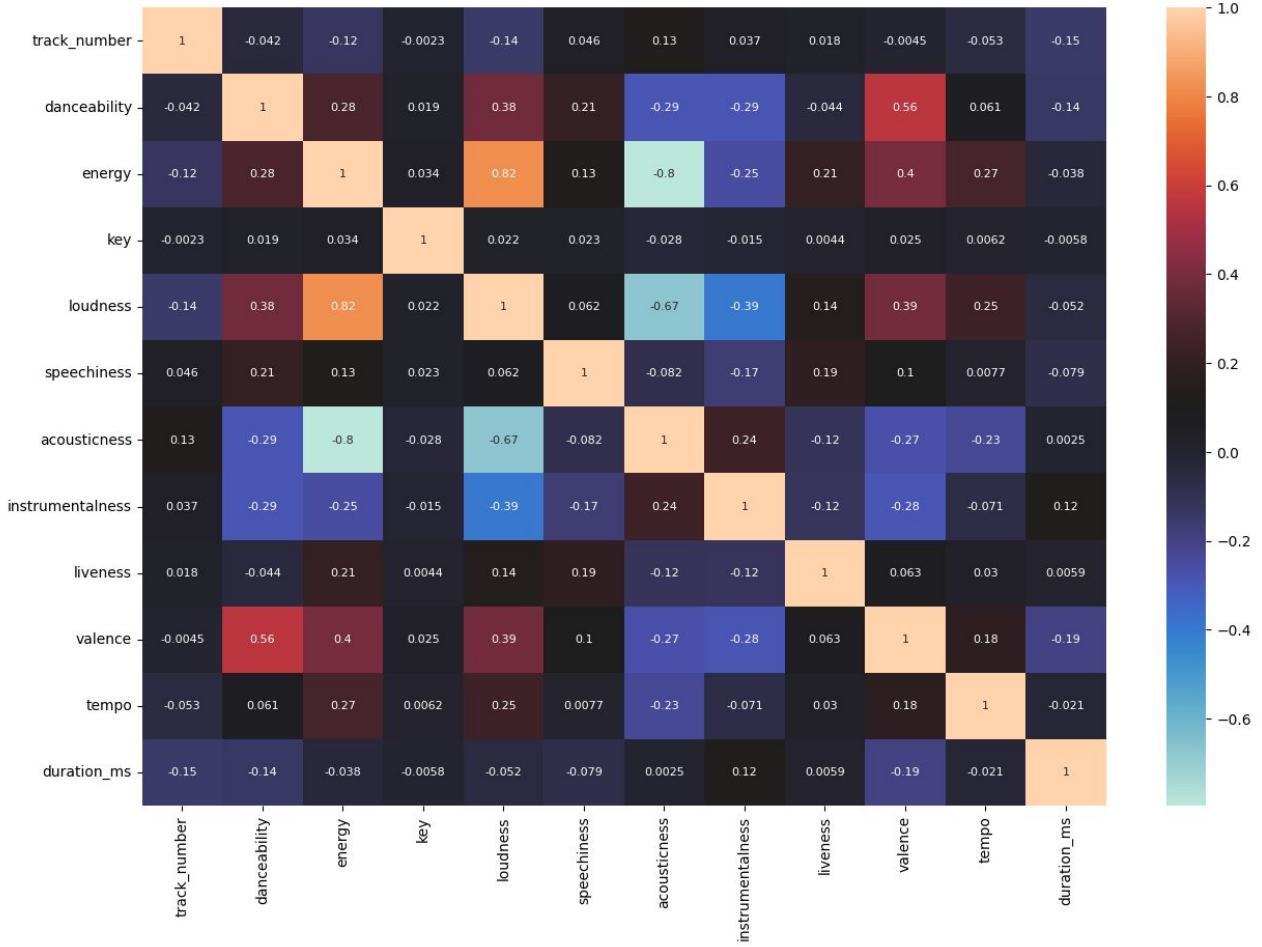


Pre-Processing



Pre-Processing

- Initially, our dataset had 24 features. However, after analyzing and preprocessing it using visualization tools such as histograms and correlation matrices, we narrowed it down to 16 features.



Methodology

We will initially focus on personalized music recommendations using content-based filtering.

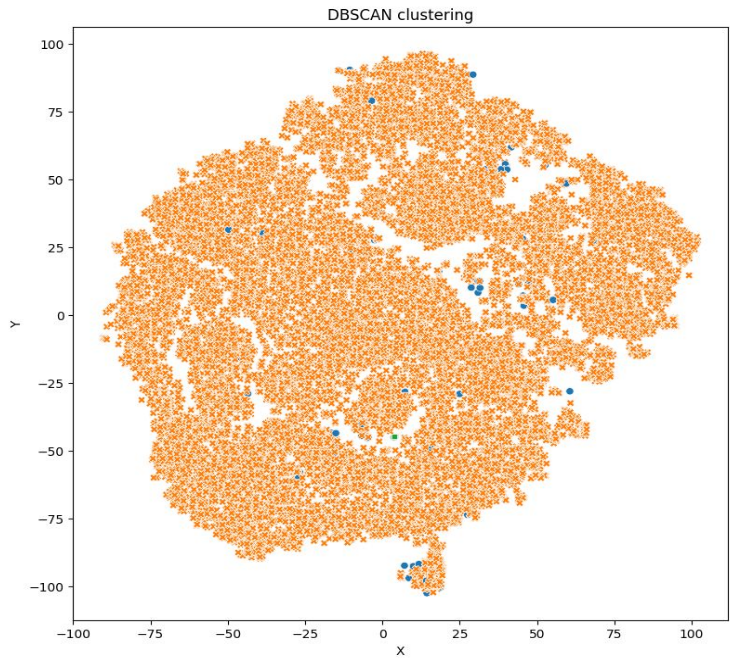
Models and Classification

We will be using 6 models DBSCAN and OPTICS which doesnt require value of k and KMeans , Birch, Agglomerative and Gaussian Mixture which requires value of k.

Models without using k number of clusters

1. DBSCAN

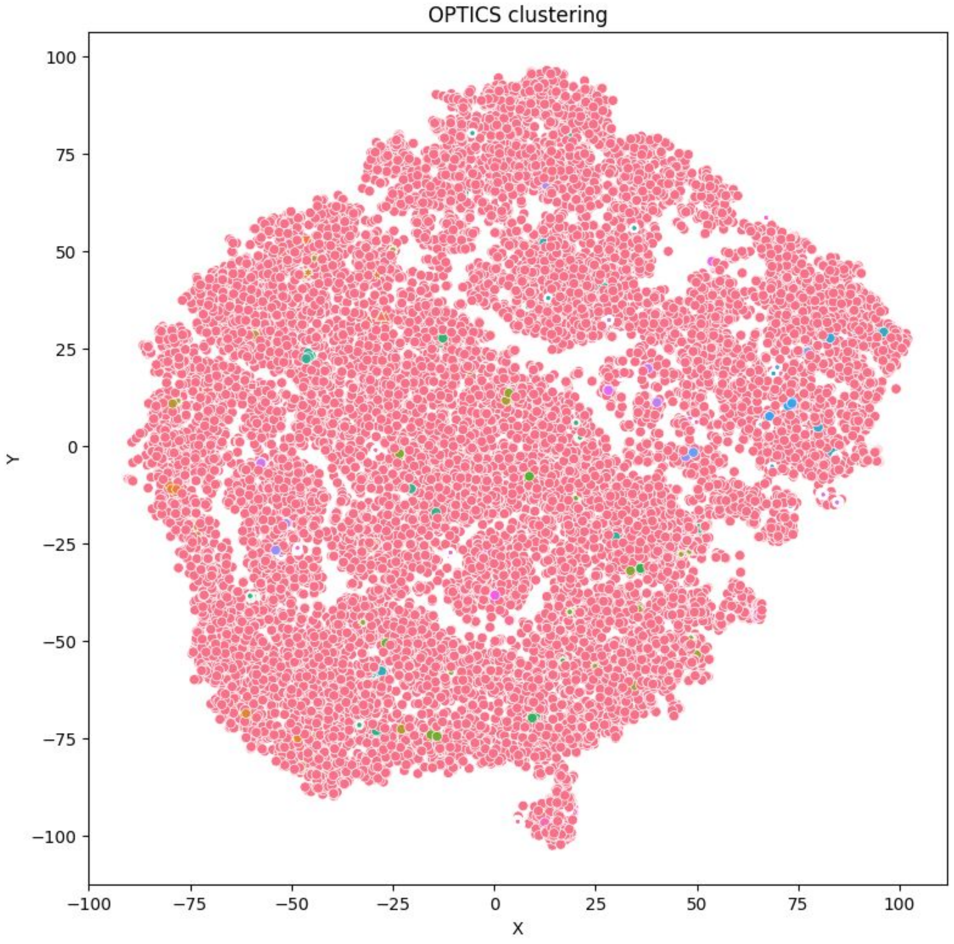
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a machine learning method used for clustering data points.
- DBSCAN groups similar tracks together by considering the density of data points. It forms clusters by identifying regions in the feature space where data points are close together.
- DBSCAN determines the number of clusters automatically based on the distribution of data points in the feature space.
- Parameters used: epsilen = 0.5; minPts = 5



Methodology

2. **OPTICS (Ordering Points to Identify the Clustering Structure):**

- A density-based clustering method.
- Creates a hierarchical representation of clusters.
- Orders data points by their density and forms clusters at different levels of density.
- Parameters include minimum points (5) and a reachability distance (0.5).



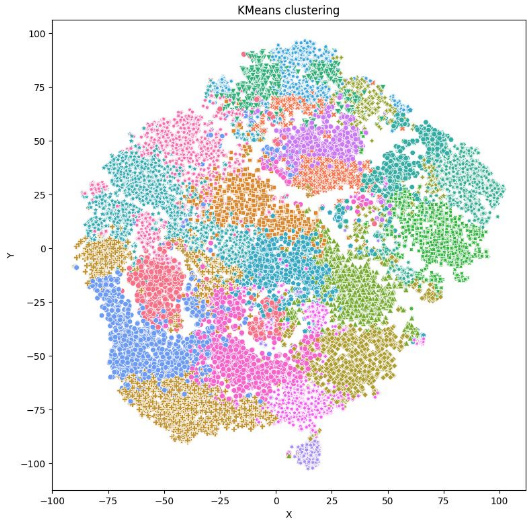
Reachability Plot

Methodology

Models with k number of clusters

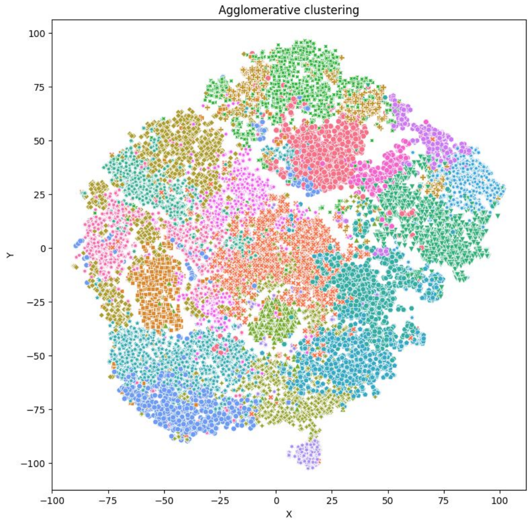
1. **KMeans**

- A centroid-based clustering method.
- Divides data into a predefined number of clusters (k).
- Iteratively assigns data points to the nearest cluster center and updates the center.
- Sensitive to the initial placement of cluster centers.
- Requires specifying the number of clusters (k).



2. **Agglomerative**

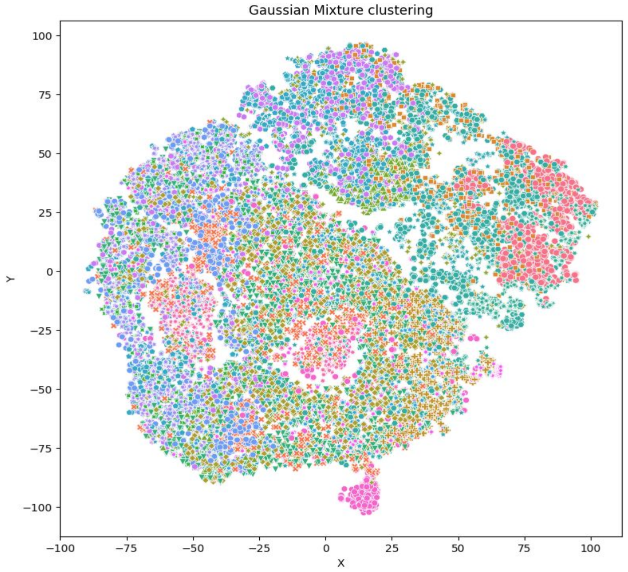
- A hierarchical clustering method.
- Starts with each data point as its cluster and iteratively merges the closest clusters.
- Forms a hierarchy of clusters (dendrogram).
- The number of clusters is determined by cutting the dendrogram at a certain level.



Methodology

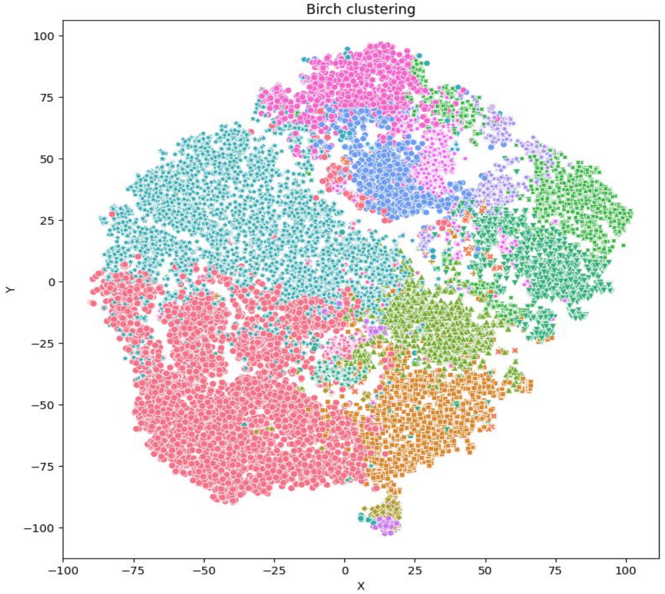
3. Gaussian Mixture:

- A probabilistic model for clustering.
- Assumes that data points are generated from a mixture of Gaussian distributions.
- It estimates the parameters (means, variances, and mixing coefficients) of these Gaussians.
- Useful for modeling clusters that may have different shapes and orientations.



4. Birch (Balanced Iterative Reducing and Clustering using Hierarchies):

- A hierarchical clustering method.
- Organizes data points into a tree structure.
- Uses a combination of clustering features and merging subclusters.
- Designed for large datasets with an emphasis on memory efficiency.



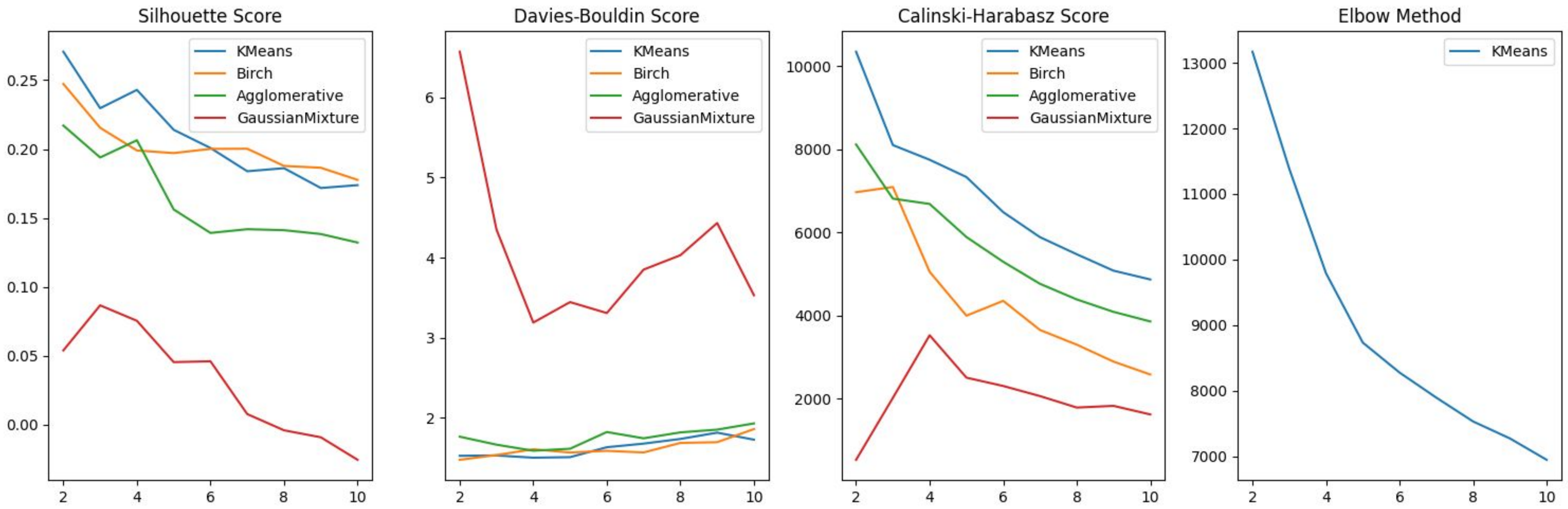
Methodology

Playlist Generation using recommendation algorithms

1. Using Cluster Similarity: Songs are represented in an N-dimensional space based on their features. By leveraging K-means clustering, songs with similar attributes are grouped together.

2. Using Feature Similarity and Content-Based Filtering: each song is represented by a series of features. The similarity between the given song and other songs in the database is determined using the cosine similarity metric. **Formula Used:** Similarity = (A.B) / (||A||.||B||)

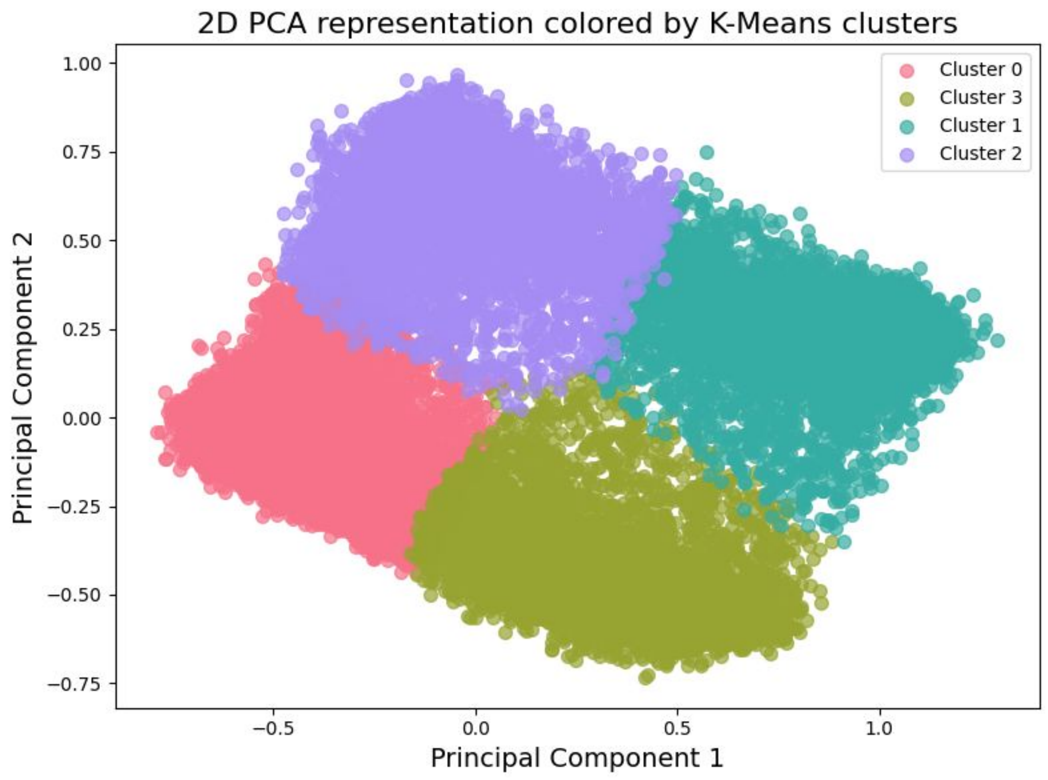
Result



Silhouette Score
Davies-Bouldin Score
Calinski-Harabasz Score
Elbow Method
KMeans
Birch
Agglomerative
GaussianMixture

Results

After examining the results from the silhouette score, Davies-Bouldin score, Calinski-Harabasz score, and the elbow test, we determined that k=4 was optimal. Based on this, we made song recommendations to users.



MUSIC RECOMMENDATION ALGORITHM (USING CLUSTER SIMILARITY)

```
given_song_features ={
    'name': 'Testify',
    'album': 'The Battle Of Los Angeles',
    'artists': ['Rage Against The Machine'],
    'track_number': 1,
    'danceability': 0.47,
    'energy': 0.978,
    'key': 7,
    'loudness': -5.399,
    'speechiness': 0.0727,
    'acousticness': 0.0261,
    'instrumentalness': 0.000011,
    'liveness': 0.356,
    'valence': 0.503,
    'tempo': 117.906,
    'duration_ms': 210133.0,
    'year': 1999.0,
    'kmeans_labels':0
}
recommended_songs = recommend(df, given_song_features, N=10)
print(recommended_songs)
```

```
['Testify', 'Guerrilla Radio', 'Calm Like a Bomb', 'Mic Check', 'Born As Ghosts', 'Maria', 'Voice of the Voiceless', 'New Millennium Homes', 'Ashes In the Fall', 'War Within a Breath']
```

MUSIC RECOMMENDATION ALGORITHM (USING FEATURE SIMILAITY AND CONTENT BASED FILTERING)

```
given_song_features ={
    'name': 'Testify',
    'album': 'The Battle Of Los Angeles',
    'artists': ['Rage Against The Machine'],
    'track_number': 1,
    'danceability': 0.47,
    'energy': 0.978,
    'key': 7,
    'loudness': -5.399,
    'speechiness': 0.0727,
    'acousticness': 0.0261,
    'instrumentalness': 0.000011,
    'liveness': 0.356,
    'valence': 0.503,
    'tempo': 117.906,
    'duration_ms': 210133.0,
    'year': 1999.0,
    'kmeans_labels':0
}
recommended_songs = music_recommend_content_based(df, given_song_features, N=10)
print(recommended_songs)
```

```
['Testify', 'Veillée spatiale', 'The Ride', "(It's Good) To Be Free - Live for Teenage Cancer Trust", 'Low Tide', "What's Wrong With You", 'Águia não come mosca', 'Seekir', 'Punks In The Beerlight', 'Appropriate Dipstick']
```

Contributions & Timeline

- **Timeline:** Slightly ahead; We are able to follow our timeline.

Contributions:

- **Aniket Kanojia** (Feature Extraction,Recommendation algorithm, optics, Birch, Agglomerative, Gaussian Mixture)
- **Ashutosh Gera** (Data PreProcessing, Feature Extraction, K means clustering,Report and Presentation)
- **Tushar Chandra** (Visualisation, analysing and improving results)
- **Piyush Kumar** (Report and Presentation, improving results)