**Introduction**

**Background**

Machine Learning and artificial intelligence have become the trend in the music streaming industry. Their potential growth were proven by the following expectations:

1. The number of users will reach the point of 1,12 billion by 2027. (Statista, n.d.)
2. The market value will be $31.4 billion by 2027. (Statista, n.d.)

Big companies, including Spotify and Apple Music, have already adapted and developed the new technology trend releasing their own innovative ML-based approaches and services such as the AI solution to recommend songs to users. Music recommendation systems predict what a user wants to listen to and answer the questions based on their learning from data such as user historical listening records. (Verma et al., 2021)

**Objective**

The goal is to understand the recommendation system, explore all possible machine learning algorithms, and solve the business and technical questions below. Through this research, the results will be informative resources for building innovative recommendation models that accelerate growth in the music-streaming industry.

1. *Which model can be applied to create the music recommendation system?*
2. *How do the models recommend songs to users?*

**Data Description**

The data set used in the project contains audio features of digital songs and song information (artist name, title, and release year) from Spotify, the music streaming company.

**Audio Features**

Audio Features are abstract characteristic that represents digital music. It is calculated from the audio signal and most features are in the range between zero crossing of the audio signal or centroids. (IGI Global., n.d.) Each audio feature has its meaning as follows. (Santos, 2017).

1. *Acousticness:* It describes how a song is acoustic in a numeric form*.*
2. *Danceability:* It shows how much a track is danceable based on musical elements.
3. *Duration\_ms:* It means the length of the track in a millisecond.
4. *Energy*: It indicates how intense and active a song is.
5. *Key:* It indicates the key of the track with values from 0 to 11.
6. *Instrumentalness:* It describes the number of vocals and instruments in a song.
7. *Liveness:* It represents the possibility that a song was recorded in a live environment.
8. *Loudness:* The overall loudness of a track in decibels, with values between -60 and 0 db.
9. *Mode:* It indicates the key in the music of the track. (1 is major, and 0 is for minor)
10. *Popularity:* It is computed on the total number of whether the music is played and how recently it is played. The value is between 0 and 100, with 100 indicating the most popular.
11. *Speechiness:* It is the degree of how many words are spoken words in a song.
12. *Tempo:* It is a speed that is calculated from the average BPM (Beats per Minute).
13. *Valence:* It shows the sentiment vibe of a song. The high valence means more positive sounds.

**Handling Missing Values, Number Scales, and Outliers**

The missing values, outliers, and different scales of numerical values may interfere with the good performance of the music recommendation algorithm. Therefore, these should be removed or handled by proper techniques.

First, there are no missing values in all columns from the given dataset throughout the investigation. Second, by checking the scales of all numeric variables, all audio features are not on the same scale. This will be solved by deploying normalization in the model-building part. Third, in *Figure 1. Box Plot of Each Audio Feature After Removing the year Column*, there are many outliers in some features. Normalization technique will be used to reduce the outliers in the data preprocessing section.

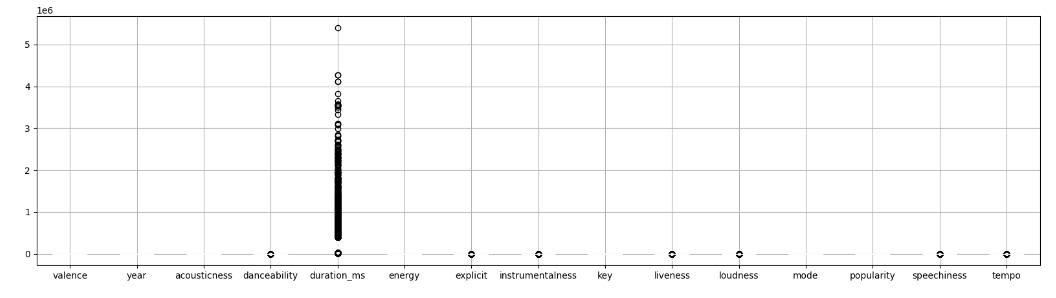


Figure 1. Box Plot of Each Audio Feature After Removing the *year* Column

**Exploring the Data**

There are 34,088 (unique) artists in the data set. Additionally, the range of year songs in the data set is between 1921 to 2020. As see *Figure 2. The Histogram of Songs by year\_group*, most songs are in the categories between the 1950s and 2010s.

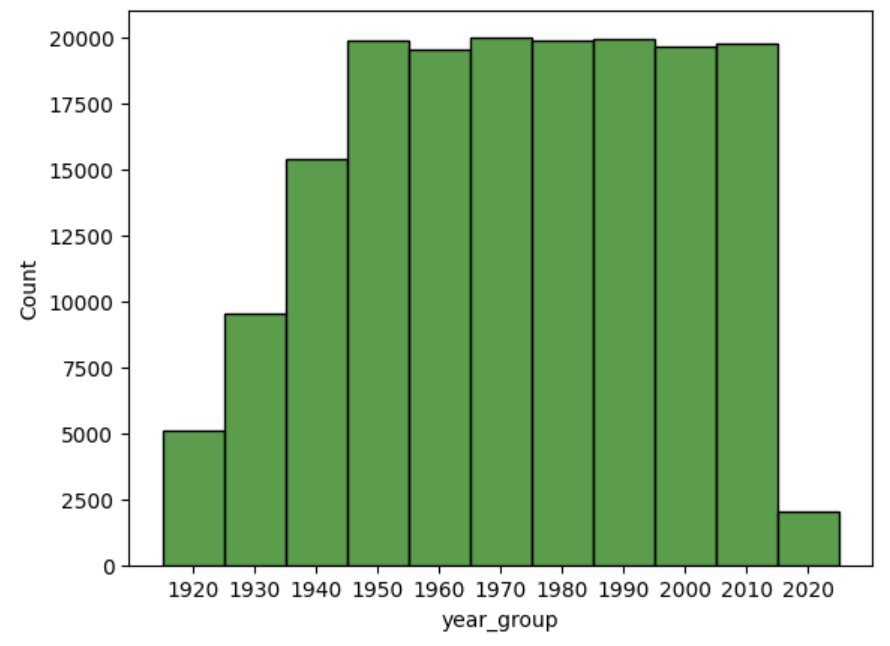


Figure 2. The Histogram of Songs by *year\_group*

**Data Modeling**

**Association Rule**

In this project, the association rule algorithm, one of the popular machine learning algorithms used to find frequent relationships, will find patterns or associations between different songs based on the pre-processed features.

**Data Pre-processing**

Discretization and one-hot encoding techniques are used in the data pre-processing for Association Rules. First, the discretization technique divides the audio features into three categories based on their percentile ranking: low (0-33%), medium (33-66%), and high (66-100%). Next, a one-hot encoding method converts the discretized data into a binary format that is compatible with the Association Rule algorithm, using the *pandas.get\_dummies()* function.

**Data Modeling**

In this step, the following functions from *mlxtend.frequent\_patterns* library are used to build the association rules among audio features: *apriori()* and *association\_rule()*. First, the *apriori()* function with the parameters (*min\_support = 0.05* and *use\_colmanes = True*) takes a set of transactions. Then, it generates a list of frequent item sets based on the minimum support threshold and the original binary column names. Second, the *association\_rule()* function discovers relationships between the variables in a dataset and finds patterns in the data that occur together more frequently than would be expected by chance with the parameters (*metric = support* and *min\_threshold = 0.5*). *Table 2. Association Rule of Audio Features of Digital Songs from Spotify* shows the first rules from the output of the association rule algorithm we run and built above.

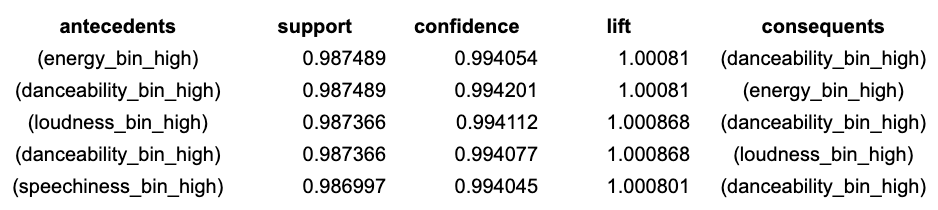


Table 2. Association Rule of Audio Features of Digital Songs from Spotify

**Evaluation**

This step is to find the best parameter to show the best association (rules) in the audio feature data set, based on the above baseline. By sorting the rules by three metrics (support, confidence, and lift), these results are compared in *Table 3. The Best Five Association Rules with Confidence, Lift, and Support.*

|  |  |  |
| --- | --- | --- |

Table 3. The Best Five Association Rules with Confidence (Left), Lift (Center), and Support (Right)

Since the goal is to identify the strongest associations among features that appear frequently, the three metrics are used together in the association rule with the following sequence as *Table 4. The Best Five Association Rules with a Combination of Confidence, Lift, and Support*.

First, confidence and lift, which indicate which item sets have the strongest association, are used. Then, support describes which item sets appear frequently in the data set. In other words, it means finding the strongest rules (associations) using confidence and lift and selecting the frequent rules in the data set.

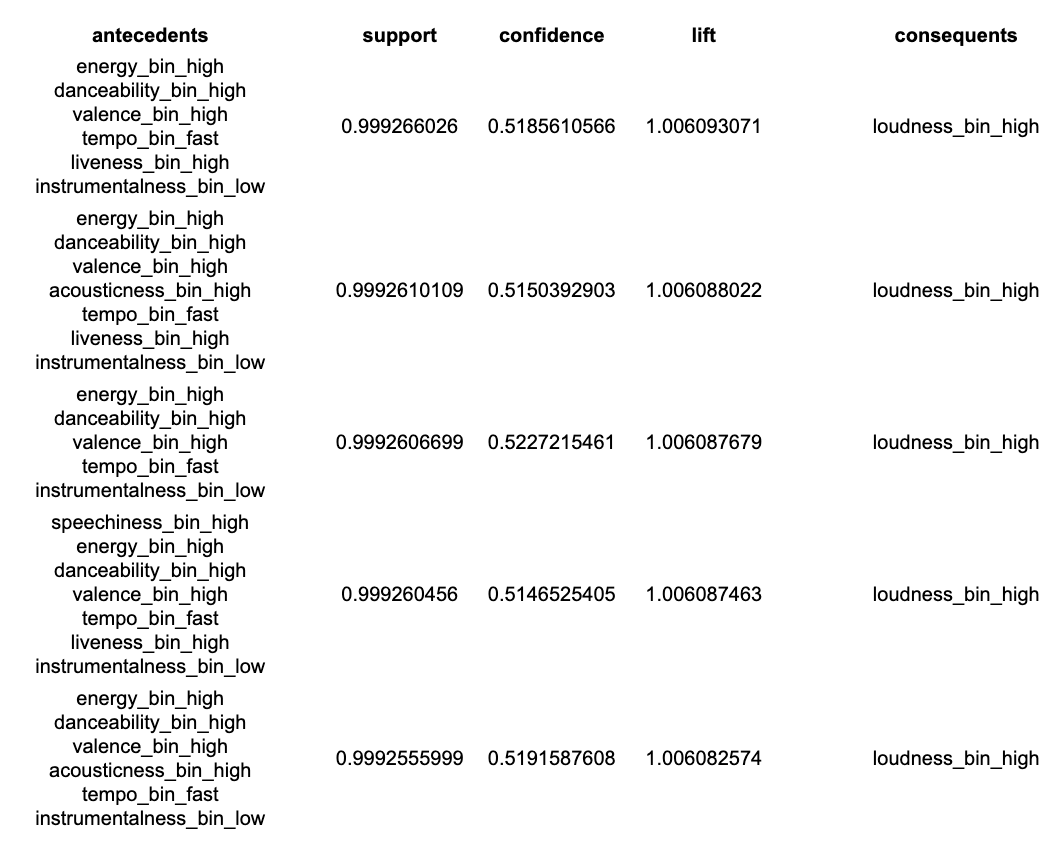


Table 4. The Best Five Association Rules with a Combination of Confidence, Lift, and Support

**Clustering**

In recommending music, the clustering model is an effective method for grouping similar songs or artists based on their shared features, even if you don’t have a historical record of listening to songs from users.

**Clustering with Content-Based Filtering**

Combining clustering models with content-based filtering can provide personalized and relevant recommendations to the user based on the similarity of the songs' features and the listening preference that they would provide.

**Data Preprocessing**

**Feature Selection**

A correlation matrix is used for feature selection, because there are many features that are highly correlated with each other so it is redundant to keep every single feature. After removing features based on the correlation matrix values, there are 6 features left: *valence, instrumentalness, key, liveness, popularity, speechiness*. *Table 5. Correlations of Selected Audio Features* describe the correlation values of each selected feature.

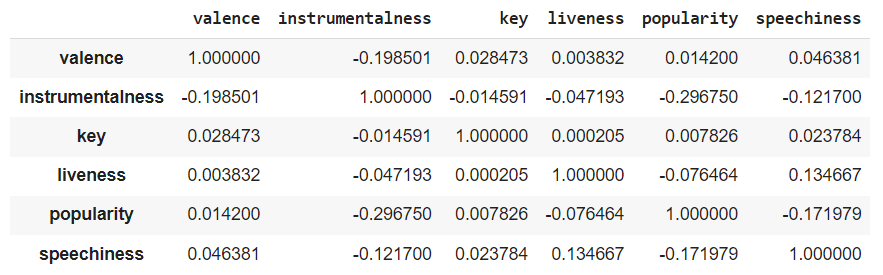


Table 5. Correlations of Selected Audio Features

**Normalization**

Normalization involves scaling the features of the dataset to a standard range to ensure that all features have equal importance in the clustering process. The MinMaxScaler normalization classifier prevents features with larger values from dominating the clustering. Figure 3. The scale of liveness, loudness, and popularity Before (Left) and After (Right) Normalization shows the results from the MinMaxScaler technique. Before normalization, the numeric variables are on different scales, but after normalization, the distribution of the features is preserved, and they are all on the same [0, 1] scale.

|  |
| --- |
|  |
|  |
| Figure 3. The Scale of *liveness*, *loudness*, and *popularity* Before (Left) and After (Right) Normalization |

**Dimensionality Reduction**

Dimensionality reduction involves reducing the number of features in the dataset while preserving as much information as possible for the low computational complexity of clustering algorithms and improvement in their accuracy. The following two commonly used dimensionality reduction techniques are used to reduce dimensions. The first technique is *PCA*, which reduces the number of features in a dataset by identifying the principal components that capture the most variance in the data. The other one is called *t-SNE*, a technique that can be used to visualize high-dimensional datasets in two or three dimensions. It works by mapping each data point to a two or three-dimensional space based on its similarity to other data points. *Figure 4. The Scatter Plot of Dimensionally Reduced Data (n\_component = 2) of Both PCA and t-SNE Techniques* describes the output from these techniques.

|  |  |
| --- | --- |

Figure 4. The Scatter Plot of Dimensionally Reduced Data (n\_component = 2) of

Both PCA (Left) and t-SNE (Right) Techniques

**Data Modeling**

**Find the Best the Number of Clustering**

Before starting to build clustering models, choosing the best number of clusters for a music recommendation can be challenging. The following method to find the best number of clusters is used. First, the *elbow method* involves plotting the within-cluster sum of squares (WSS) against the number of clusters and looking for the "elbow" point where the rate of decrease in WSS starts to level off. Second, *the Silhouette score* is a metric that measures how similar an object is to its own cluster compared to other clusters. A higher silhouette score indicates better cluster separation. Through the method, the best numbers of clusters from the two methods are defined as 10 in *Figure 5. The Plots of Number of Clusters vs. WCSS and Silhouette Score*.

|  |  |
| --- | --- |

Figure 5. The Plots of Number of Clusters vs. WCSS and Silhouette Score

**Model Training**

In the model training section, the following five clustering models are used to find the best algorithm to create clusters among the songs, based on their audio features: K-Means, DBSCAN, Hierarchical & Agglomerative, and Spectral Clustering. Note that the number of clusters is 5 for every clustering model based on the number of audio features and results from elbow and silhouette methods. The results are described in *Figure 6. The Visualization of Each Clustering Model.*

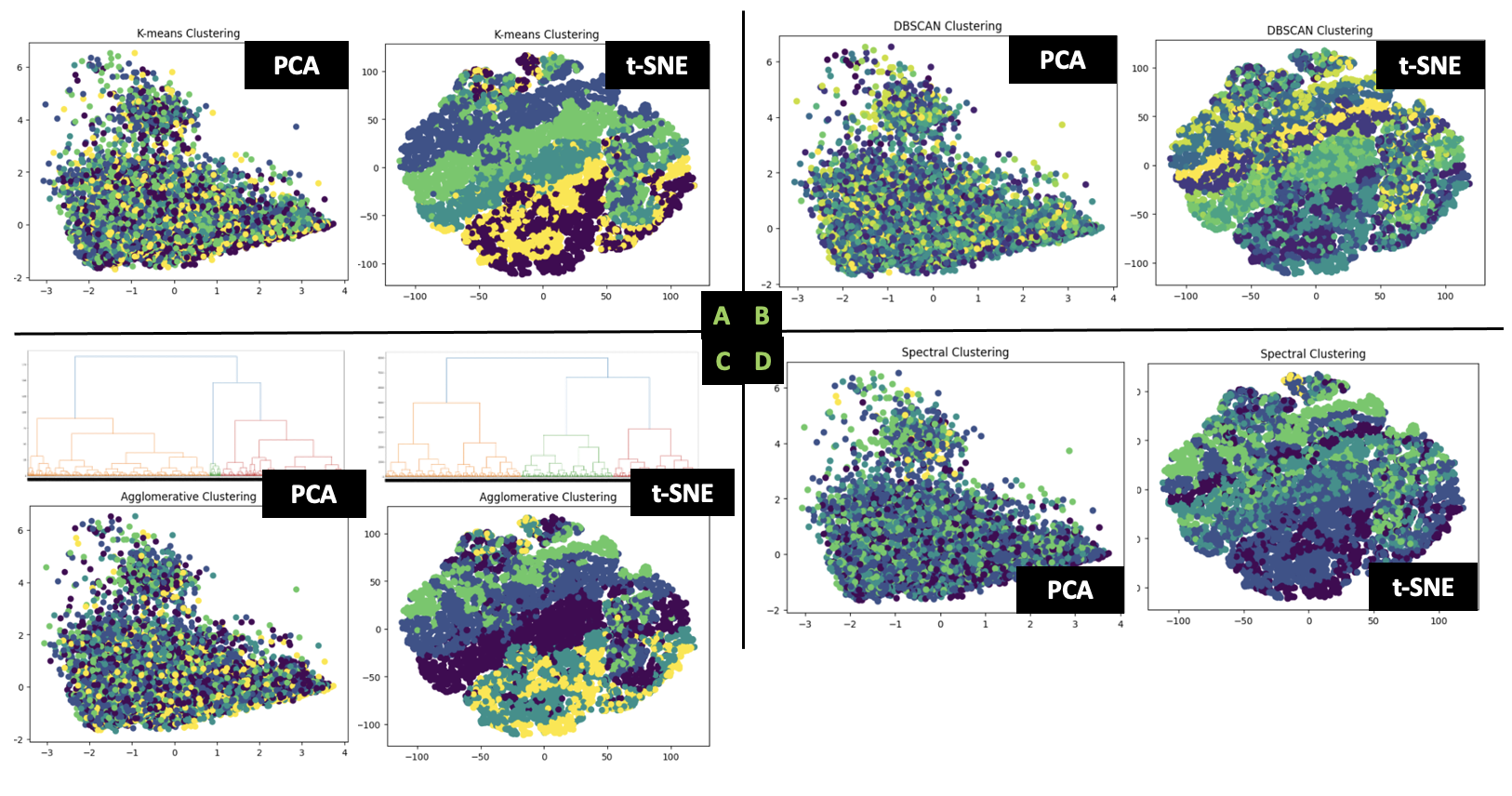
****

Figure 6. The Visualization of Each Clustering Model

**Evaluation**

The evaluation of clustering models is an important step in assessing the quality of the clusters produced by the algorithm. Metrics like Silhouette Score and Calinski-Harabasz Score can be used to evaluate the compactness and separation of the clusters.

**Silhouette Score and Calinski-Harabasz Score**

The *Silhouette Score* measures how well each data point fits into its assigned cluster, and how distinct it is from the other clusters, while the Calinski-Harabasz Score measures the ratio of between-cluster variance to within-cluster variance. *Figure 7. The Silhouette Scores and Calinski-Harabasz Scores of Clustering Models* show the scores of each clustering model that are built above. Based on the results, K-means will be the best model to build clustering models for the following reasons. First, its average silhouette score is 0.503, which means the data point in k-mean clusters is very well matched to its own cluster, but not to neighboring clusters. Second, the Calinski-Harabasz Score is 74,711.500, meaning that the K-means model shows the better clustering results.

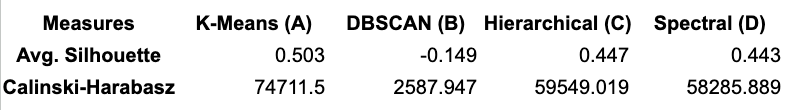
****

Figure7. The Silhouette Scores and Calinski-Harabasz Scores of Clustering Models

**Results**

1. **Association Rules**

* To find the best association rule, support will be used as a metric with a minimum value of 0.05 and a minimum threshold of 0.5.
* Based on the above-defined rules, it is possible to interpret that most active digital songs contain instrumental-based features in the frequent item sets.

1. **Clustering**

* Based on the above study, the best model for music recommendation systems is K-means clustering.
* This is because it shows the highest performance of the matching data points in its own clusters and overall, better clustering results.

**Further Studies**

**Applications**

The following figures describe the recommended songs from music recommendation functions (system), written in Python, based on the association rule and clustering models that are built in the above study with the input value (The artist name is Eminem, who is a famous American hip hop artist.). Even though there are no exact evaluation standards (users’ listening record or preference) for the recommendation, it is possible to conclude that the left-side recommended songs are more acceptable for most people.

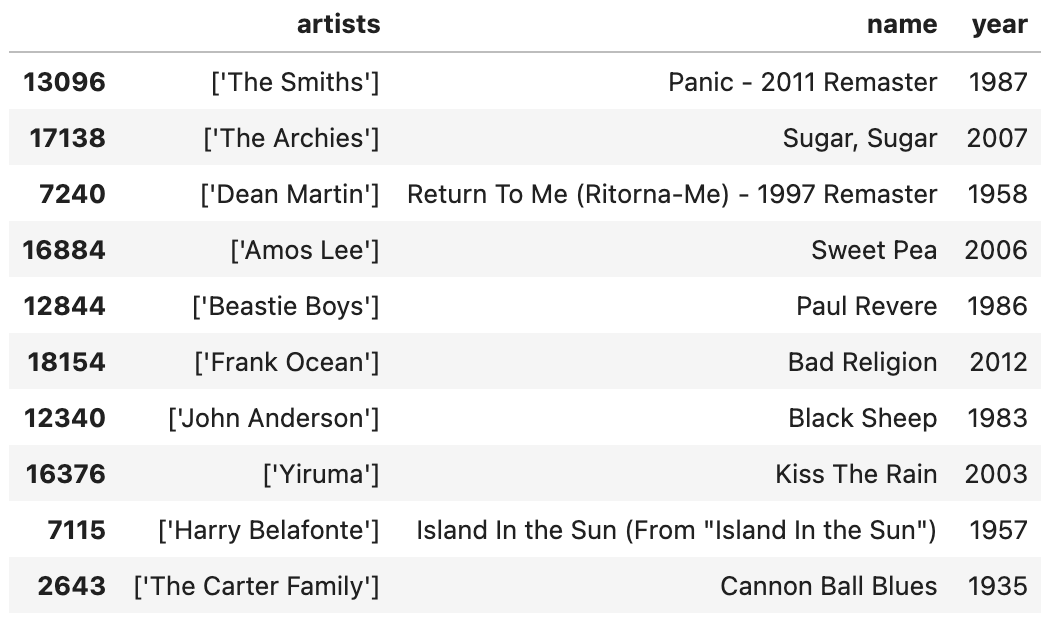
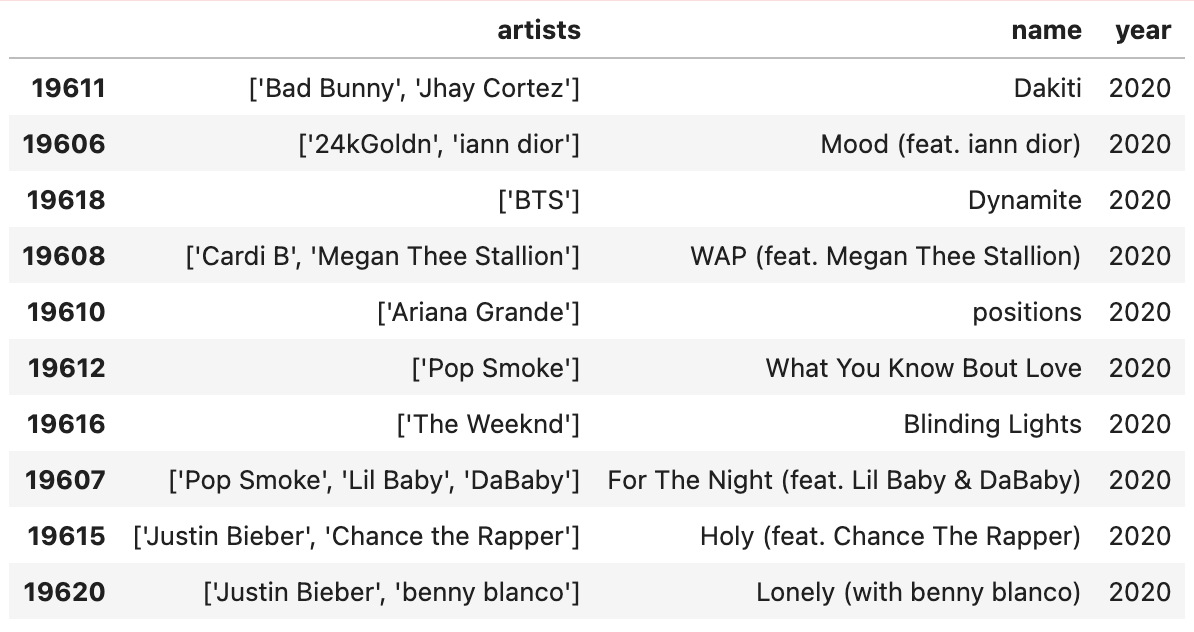
****

Figure 8. The List of Recommended Songs from Association Rule (Left) and Clustering (Right) Model

**Improvements (Suggestions for Further Studies)**

To evaluate and improve the music recommendation system, we also need user data sets (personal information or listening history) to conduct the reinforcement learning for ranking the recommended songs and presenting the best.

**Reference**

Jolliffe, I. T. (2011). Principal component analysis. Springer.

Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9(Nov), 2579-2605.

*Music streaming - worldwide: Statista market forecast*. Statista. (n.d.). Retrieved April 2, 2023, from https://www.statista.com/outlook/dmo/digital-media/digital-music/music-streaming/worldwide#:~:text=Revenue%20in%20the%20Music%20Streaming,US%2425.71bn%20in%202023.

Müller, A. C., & Guido, S. (2016). Introduction to Machine Learning with Python. O'Reilly Media, Inc. (pp. 444-445).

Plantinga, B. (2018, April 29). What do Spotify's audio features tell us about this year's Eurovision Song Contest? 🤔. Medium. Retrieved April 22, 2023, from https://medium.com/@boplantinga/what-do-spotifys-audio-features-tell-us-about-this-year-s-eurovision-song-contest-66ad188e112a

RPubs. (n.d.). Retrieved April 22, 2023, from https://rpubs.com/kdrdatascience/862303

*What is audio features*. IGI Global. (n.d.). Retrieved April 2, 2023, from https://www.igi-global.com/dictionary/audio-features/1810#:~:text=1.,defined%20in%20the%20MPEG7%20standard).

Santos, J. D. D. (2017, May 31). *Is my Spotify music boring? an analysis involving music, data, and machine learning*. Medium. Retrieved April 2, 2023, from <https://towardsdatascience.com/is-my-spotify-music-boring-an-analysis-involving-music-data-and-machine-learning-47550ae931de>

Scikit-learn. (n.d.). Clustering performance evaluation. Retrieved September 28, 2021, from <https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation>

Santos, J. D. D. (2017, May 31). Is my Spotify music boring? an analysis involving music, data, and machine learning. Medium. Retrieved April 2, 2023, from <https://towardsdatascience.com/is-my-spotify-music-boring-an-analysis-involving-music-data-and-machine-learning-47550ae931de>

Tan, P. N., Steinbach, M., & Kumar, V. (2006). Introduction to data mining. Addison-Wesley Longman Publishing Co., Inc.