Music Popularity Predictor

Sam Springer
Computer Science
University of Colorado Boulder
sasp7990@colorado.edu

Sean McCormick
Computer Science
University of Colorado Boulder
semc2429@colorado.edu

Jenna Dean
Computer Science
University of Colorado Boulder
jede4828@colorado.edu

Ryan Power
Computer Science
University of Colorado Boulder
ryan.power@colorado.edu

ABSTRACT

We have created models that provide song recommendations and that predicate artist and song popularity. In particular, the models predict artist and song popularity based on attributes of the artist and the song. We collected the attributes from the Spotify API and, in turn, utilized statistical predictive analysis approaches to derive the attributes to discover patterns between the attributes and corresponding popularity. For instance, we used the Apriori algorithm to generate attributes that appear frequently with one another and correspond to popularity of an artist or song. This allowed us to create predictive music popularity models for song and artist popularity. In addition, we also used statistical predictive analysis approaches to generate song and artist recommendations.

INTRODUCTION

This day and age, Spotify, Pandora, and Apple Music have a limitless selection of streaming music that gives listeners the ability to easily and quickly connect with their favorite artists and songs--by just a click of a button. These platforms also generate music recommendations based on the likes and dislikes of each listener. For example, Apple Music generates a playlist of roughly 25 songs each week for a listener based on the type of music the listener listened to the previous week.

Beyond the ingenuity these platforms offer to listeners, these platforms likewise help the music industry understand what listeners prefer, and, as such, create music that is catered to these preferences. In particular, Spotify offers a web API that allows developers to collect various attributes, such as energy, tempo, danceability, of a song. Given the ability to harness these attributes from Spotify, we plan to create several models that can successfully predict the popularity of a song or artist based on the quantitative and qualitative characteristics obtained from a song and/or artist. These models aim to perform the following: (1) predict the popularity of a song or artist based on correlating song or artist attributes to those of popular song or artist attributes, respectively; (2) discover attributes that are correlated with popular songs or artists; (3) predict popularity of a song or artist based on genres that are popular.

Implementation of these models will be of great benefit to the music industry. It will assist the music industry in better understanding what attributes, e.g., danceability, tempo, energy, it should focus on when creating a new song or even when signing a new artist.

LITERATURE SURVEY

In the late 1990s, music platforms allowed music discovery through music recommendations, ratings, etc., based on the likes of other listeners. Pandora and the Music Genome Project were the impetus of taking a more analytical approach to music discovery by analyzing the structure of a song so as to discover similar songs that a listener might like. [1]. Following the Music Genome Project launch, many researchers have adopted this analytic approach of analyzing the characteristics of music to provide listeners with songs that are similar to the characteristics of music they have previously listened to. For example, Elbir and Aydin put forth a recommendation engine that classifies music based on acoustic characteristics, which in turn recommends music having similar acoustics to listeners. [2]. A number of other recommendation engines, such as Li and Tzanetakis [3] and Holzapfel and Stylianou [4], also use acoustic characteristics to recommend music to listeners.

PROPOSED WORK

For data collection, we plan to use several datasets available on Kaggle[1] as well as data collected from the Spotify API. We will combine the data and use the Pandas Python library to transform the JSON formatted data from the API into easily manipulated and searchable DataFrames. To ensure data quality, we will check for completeness by using attribute means or regression to fill in any missing values where appropriate or deleting the data object if necessary. We will also search for and eliminate all duplicate data and outliers, such as artists and songs with a popularity value of 0. To reduce dimensionality and work only with data that will provide meaningful insights, we will delete those attributes highly correlated with other attributes. To derive data and discover patterns, we plan to use k-means clustering which will allow us to group related data into distinct clusters, and the Apriori algorithm to generate frequent item sets, as well as other statistical analysis approaches to discern the relationship between artist/song attributes and popularity.

Our process for predicting artist and song popularity will differ from prior work done by organizations such as Pandora and its Music Genome Project in that we will not be recommending artists based on users' listening histories. Unlike The Music Genome Project, whose predictive method "responds to each individual's tastes [5]", we will be attempting to predict overall popularity through song characteristics alone. Our process for determining popularity will also differ from Spotify's, which they describe as being "based, in the most part, on the total number of plays the track has had and how recent those plays are [6]." Despite these differences, we will use Spotify's popularity index as a gauge to evaluate our results.

DATA SET

Our dataset consists of fifteen attributes and 28,680 data objects, each of which represents a unique artist. The dataset contains multiple characteristics for every artist available for streaming on Spotify, such as tempo and song length, which were obtained through averaging the song characteristics of an artist's entire body of work. Thirteen of the attributes are numeric, while the remaining two, mode and genre, are binary and nominal, respectively. The mode attribute is asymmetric and skewed towards major with the majority of songs written in major scales. Eleven of the thirteen numeric attributes are floats, while the last two attributes are represented by integers. Among the numeric attributes, the majority are continuous, ratio-scaled and normalized between 0 and 1. In addition to the dataset described above, we will also be querying the Spotify API in order to obtain more detailed and song specific information such as release date, individual song characteristics, and song popularity. This will be used to supplement our dataset and will allow us to ask and answer meaningful questions about both artists and songs.

[1] https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks

EVALUATION METHODS

Once the data is generated, cleansed and analyzed, our goal is to utilize the song attributes to predict artist and song popularity. We will evaluate our success by comparing the songs and artists that we deem as popular to the ones that are considered popular by Spotify. As mentioned above, Spotify's popularity is "based, in the most part, on the total number of plays the track has had and how recent those plays are", so we will compare our songs and artists to the Spotify algorithm to determine whether our analysis was accurate [6].

TOOLS

There are many tools that we plan to use for this project. We will use both Kaggle and Spotify to supply our data. Kaggle is a machine learning and data science community where users can publish, explore and build data models. As stated above, the Kaggle dataset contains a multitude of attributes and objects. Alongside the Kaggle dataset, we plan to use the Spotify API in conjunction with a Python script to generate song/artist attributes and features. Once the data is generated, we will put it into DataFrames in order to analyze and manipulate it.

Once we have established our formatted dataset, we plan to use NumPy, Pandas and mlxtend - Apriori Python libraries to begin our analysis. The NumPy and Pandas libraries will assist in performing mathematical operations (such as minimum, maximum, average, etc.) on the attributes, and we will also use them to generate charts on the fly. These charts will be useful in determining whether two attributes are correlated and viewing the max, min, quantiles, etc., within the various attributes. The mlxtend - Apriori will be used to generate frequent item sets to show general trends within the data.

Lastly, we plan to use Tableau. Tableau is a software used to create dataset visualizations. Our hope is that we can cleanse the data using Python and then upload the cleansed data via a CSV file into Tableau. Once the data is in Tableau, we will create charts and graphs where users can easily view the song and artist popularity based on song attributes.

TECHNIQUES APPLIED:

There are many techniques that were applied to clean, preprocess, classify, and cluster the data in order to analyze it. These techniques helped the team come to conclusions about which attributes were most indicative of popularity.

As far as data cleaning and preprocessing is concerned, we used a number of techniques. First off, we used normalization techniques to apply k-means and Naive Bayesian methods. For both these techniques, we normalized the following attributes: Time Signature, Tempo, Key, Duration (ms), and Loudness. We set each of these attributes to between 0 and 1 for easier comparison and analysis. Secondly, we cleansed the date fields. Dates came in many different formats including MM/DD/YYYY and YYYY, so we modified these to only show the year. We also cleansed all outliers. For example, we removed data that had a date of 1900 and also removed tracks with a popularity equal to 0. From there, we modified the popularity attribute to make it binary (i.e., either 0 or 1). We set all popularity values that were originally greater than or equal to 50 to 1, and we set all popularity values that were less than 50 to 0. Lastly, for Naive Bayes, we filtered the data, so it included data from 2020-2021. By doing so, we were able to see how song characteristics impacted popularity without influence from release date.

Once the data was cleaned, we used classification and clustering techniques to further analyze the data. Our techniques included, K Nearest Neighbors, Random Forest, Logistic Regression, Naive Bayes, K-means Clustering, and Apriori.

For Naive Bayes, we split data into testing and training groups. We used 75% of the data to train the algorithm and 25% to test.

Our second model used was the Gaussian Naive Bayes to predict popularity based on tempo, loudness, explicitness, energy, danceability, valence, acousticness, and instrumentalness. For this model, our accuracy ranged from 63%-68%, our precision ranged from 63%-68%, and our recall ranged from 80%-93%.

Our K-means algorithm looked a bit different from Naive Bayes. For this algorithm, we chose to create 9 clusters to find which attributes were most indicative of popularity.

For our regression analysis, we used multiple linear regression with 15 attributes including Release Date, Danceability, Energy, Key, Loudness, Mode, Speechiness, Acousticness, Instrumentalness, Liveliness, Valence, Tempo, Time Signature, Duration (ms), and Explicit. Our logistic regression popularity constraints included analyzing data with a popularity greater than 20, song duration that was greater than or equal to 120 seconds and less than or equal to 300 seconds. For this algorithm, we analyzed songs that were released between 2019 and 2021, which included 22,440 tracks. This algorithm yielded an accuracy of about 69%, a precision of about 69%, and a recall of about 99%.

Random Forest popularity prediction was done using the same constraints as logarithmic regression, but this algorithm yielded an accuracy of about 68%, a precision of about 69%, and a recall of about 99%.
K-Nearest Neighbor's accuracy and precision were similar to Random Forest, but the recall was about 93%.

The final classification method that we used was Apriori. To use the Apriori algorithm, we included the following data points: Artist Name, Number of Complete Recommendations, Number of Recommendations per Complete Recommendation, and the Minimum Support. From there, we used the Apriori algorithm to find frequently occurring genres associated with popular artists (popularity >=60).

RESULTS:
The project demonstrated that the predictive popular music models have the possibility of predicting popularity based upon song and artist attributes.
We used the following methods to create predictive music popularity models to predict the popularity of a song or artist: logistic regression; Random Forest; K Nearest Neighbors; Gaussian Naive Bayes; and Apriori.

With the predictive music popularity models, we found acoustic correlations of songs captured from the Spotify API for the music genres rock and country as well as for top artists and tracks (a list of songs). Figures 1-4 illustrate the acoustic characteristics that we will base our predictive models from. The higher correlation number (illustrated with darker colors) indicates a strong correlation between the acoustic attributes and popularity. Particularly, figures 1-4 show a high correlation value between energy and loudness. Thus, the correlation between loudness and energy yields the greatest accuracy when using our predictive models to predict the popularity of songs and artists.

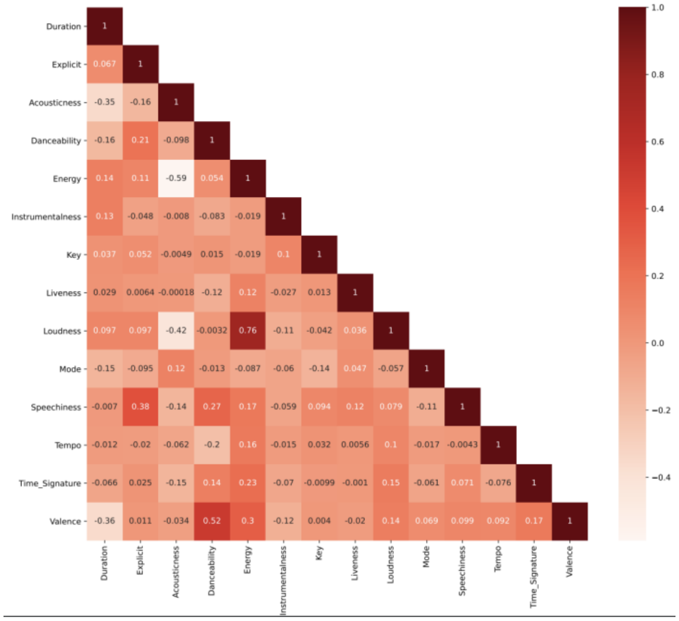


FIGURE 1. Graph depicting the acoustic attribute correlation for Rock.

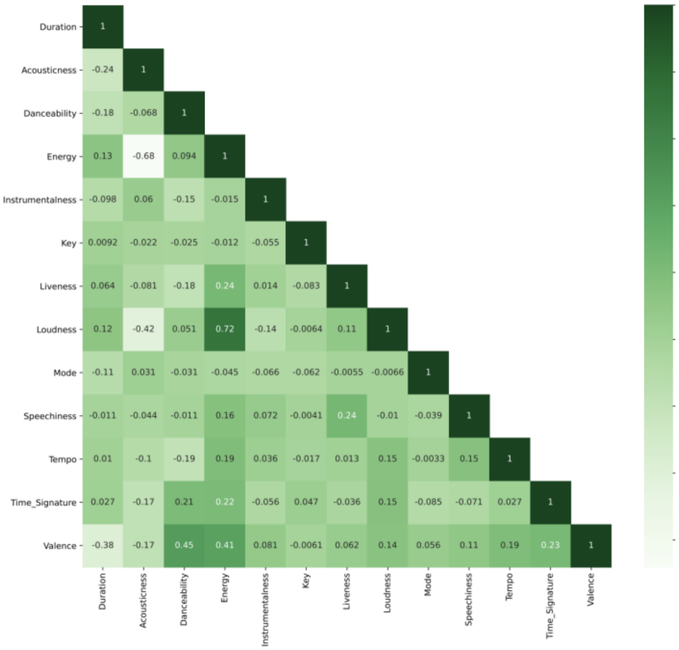


FIGURE 2. Graph depicting the acoustic attribute correlation for Country.

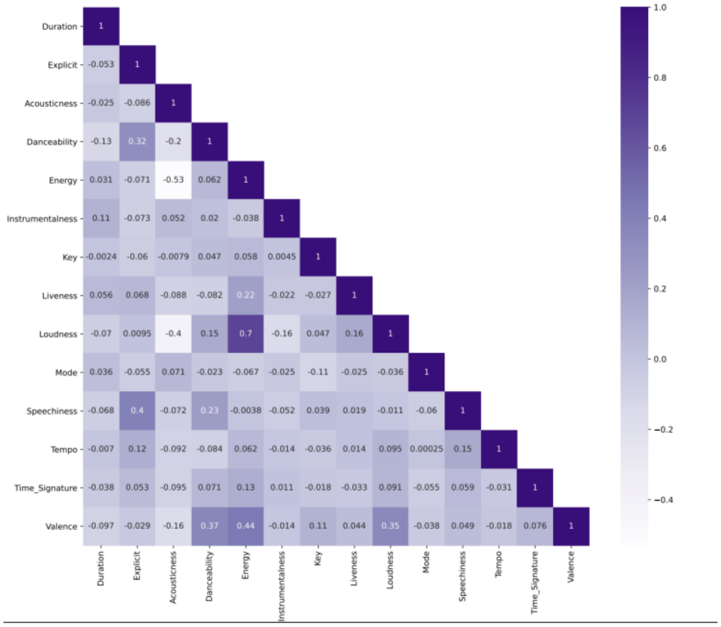


FIGURE 3. Graph depicting the acoustic attribute correlation for Top Artists.

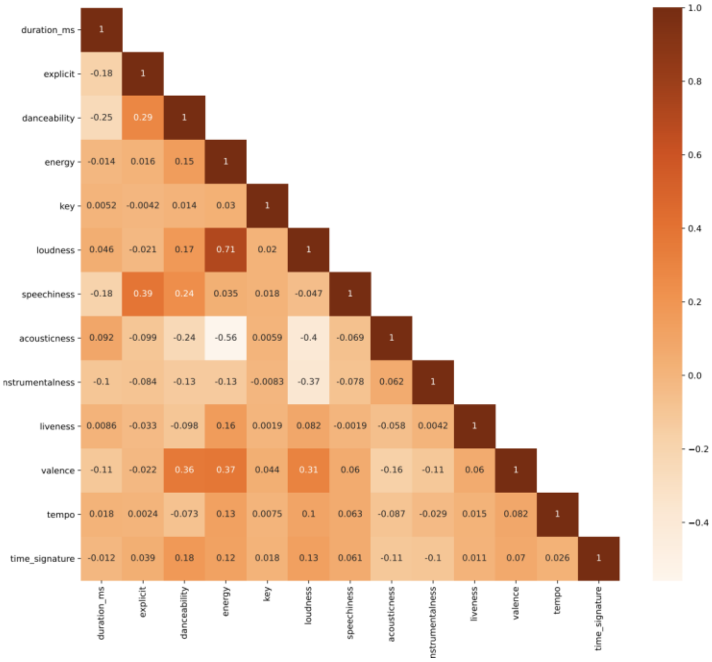


FIGURE 4. Graph depicting the acoustic attribute correlation for Tracks.

Furthermore, using Naive Bayes algorithm we created a predictive music popularity model. After creating the model, we then created 10,000 songs with the attributes of energy, loudness, explicit, and danceability. Figure 5 reproduced below shows the average value of each attribute for a popular song predicted by the model. The model demonstrates that a song that contains the foregoing attributes and corresponding values is predicted to be popular.

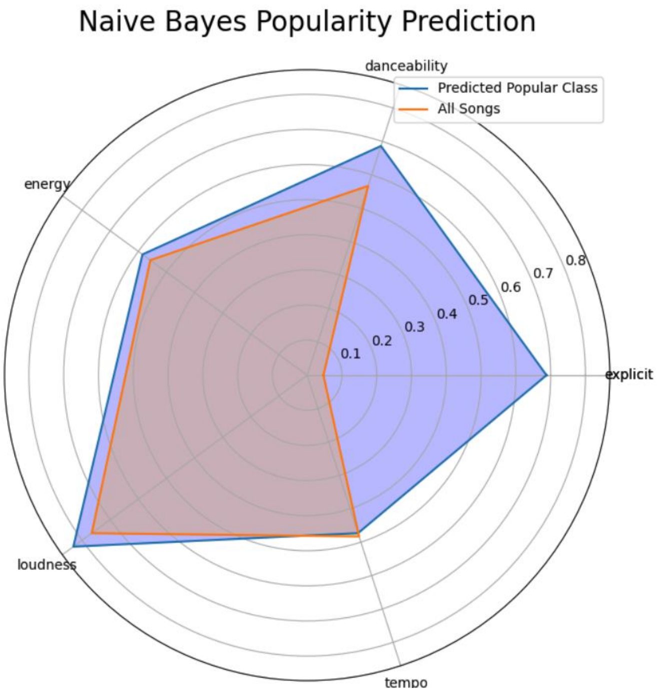


FIGURE 5. Naive Bayes Predictive Music Popularity Model shows that energy, loudness, explicit, and danceability is correlated with popularity.

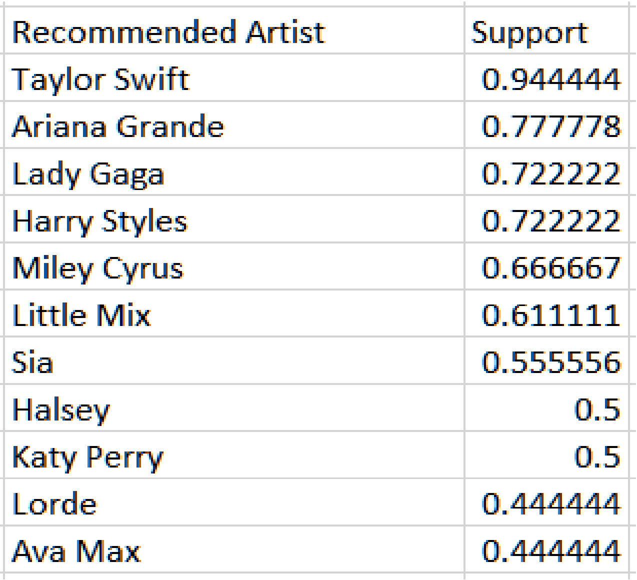
The Apriori algorithm was used in multiple ways to mine meaningful knowledge from our datasets. One method involved querying the Spotify API for artist and song recommendations. The sole input of the Python script that was written to query the Spotify API and collect and parse the returned set of recommendations was a user's favorite artist. Other parameters were present in the script however to modify returned results. A basic example of how this Python scripts executed is as follows:

1. Specify the artist of choice.
2. In the Python script, select the number of recommendation sets returned, the number of recommendations per recommendation set, and the minimum support.
3. An illustration of step 2 would be as follows:
   a. Recommendations Sets received from the Spotify API after a query is sent.

      i. a1, a2, a5
      ii. a2, a3, a4
      iii. a1, a3, a5

4. The above table shows three recommendation sets returned from the Spotify API with three recommendations received per set. Minimum support remains unchanged at 0.1.

Given the computation resources at the group's disposal, these parameters were set to higher values to acquire more meaningful data. The results of one complete execution of this scripts with the input being one of 2021's top artists, Dua Lipa, is shown below:



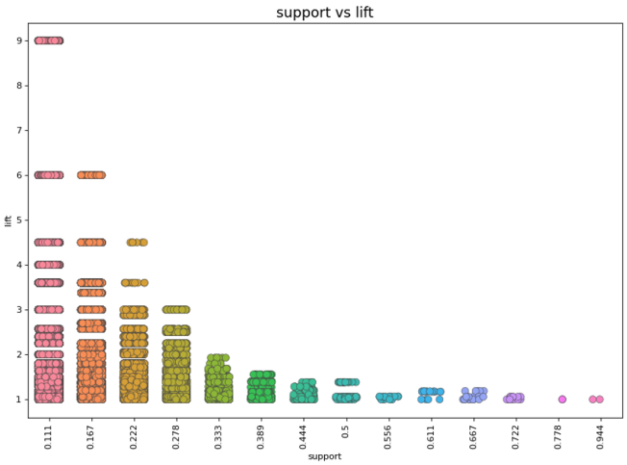
Recommended Artist	Support
Taylor Swift	0.944444
Ariana Grande	0.777778
Lady Gaga	0.722222
Harry Styles	0.722222
Miley Cyrus	0.666667
Little Mix	0.611111
Sia	0.555556
Halsey	0.5
Katy Perry	0.5
Lorde	0.444444
Ava Max	0.444444

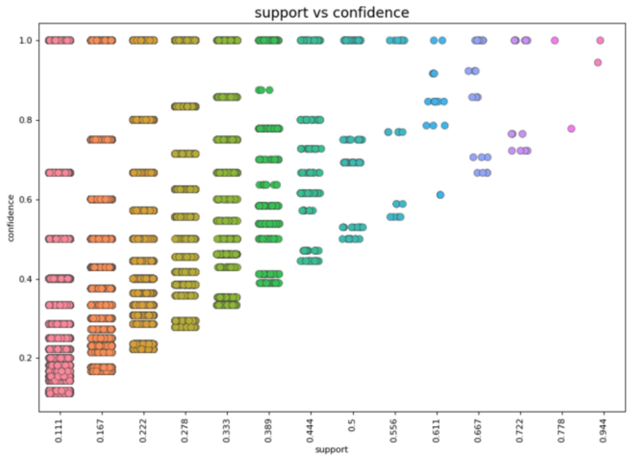
Artists recommended for you:

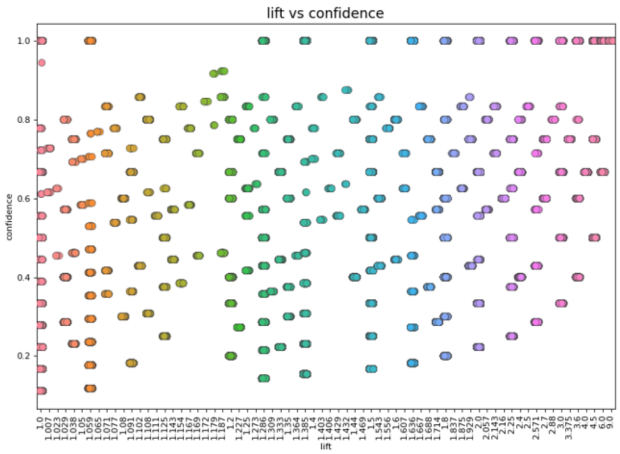
1. Taylor Swift
2. Ariana Grande
3. Lady Gaga
4. Harry Styles
5. Miley Cyrus

Other artists you may be interested in:

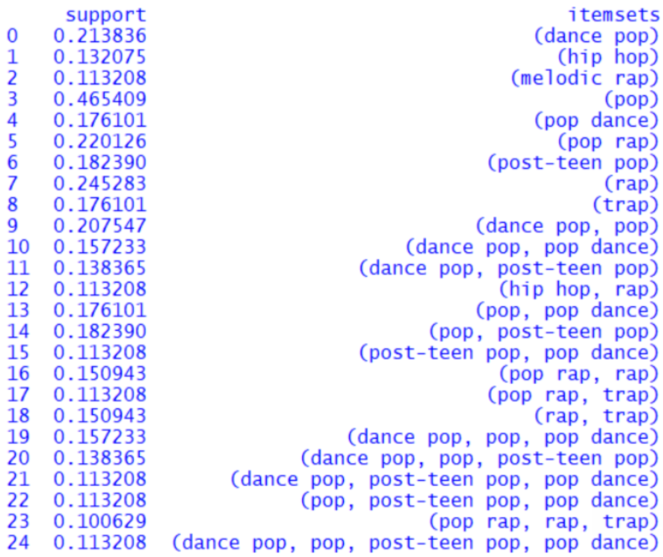
1. Little Mix
2. Sia
3. Halsey
4. Katy Perry
5. Lorde
6. Ava Max







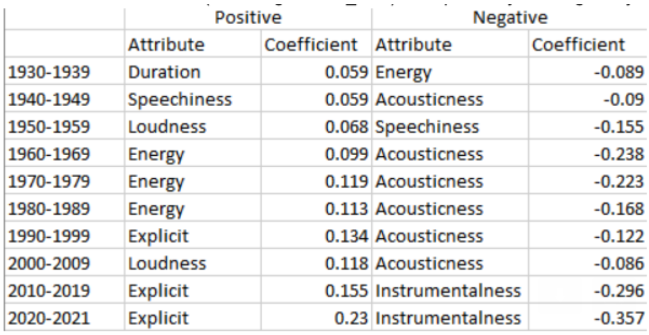
A second use of the Apriori algorithm was to find frequently occurring genres associated with popular artists (popularity >=60). Particularly, dance, pop, and rap had the highest support as shown in Figure 6 below.



```
     support                                       itemsets
0   0.213836                                    (dance pop)
1   0.132075                                      (hip hop)
2   0.113208                                  (melodic rap)
3   0.465409                                          (pop)
4   0.176101                                    (pop dance)
5   0.220126                                      (pop rap)
6   0.182390                               (post-teen pop)
7   0.245283                                          (rap)
8   0.176101                                         (trap)
9   0.207547                               (dance pop, pop)
10  0.157233                         (dance pop, pop dance)
11  0.138365                     (dance pop, post-teen pop)
12  0.113208                                (hip hop, rap)
13  0.176101                               (pop, pop dance)
14  0.182390                           (pop, post-teen pop)
15  0.113208                     (post-teen pop, pop dance)
16  0.150943                                (pop rap, rap)
17  0.113208                               (pop rap, trap)
18  0.150943                                   (rap, trap)
19  0.157233                    (dance pop, pop, pop dance)
20  0.138365                (dance pop, pop, post-teen pop)
21  0.113208          (dance pop, post-teen pop, pop dance)
22  0.113208                (pop, post-teen pop, pop dance)
23  0.100629                          (pop rap, rap, trap)
24  0.113208     (dance pop, pop, post-teen pop, pop dance)
```

FIGURE 6. support values for genre itemsets of popular artists.

Performed R-squared for multiple linear regression using as independent variables the 15 song attributes (i.e., release date, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, time_signature, duration_ms, and explicit) and popularity as the dependent variable, we found R-squared value 0.329 (rounded to the nearest hundredth). Furthermore, Table 1 below shows the most positively and negatively attributes correlated with popularity for each decade.



	Positive		Negative	
	Attribute	Coefficient	Attribute	Coefficient
1930-1939	Duration	0.059	Energy	-0.089
1940-1949	Speechiness	0.059	Acousticness	-0.09
1950-1959	Loudness	0.068	Speechiness	-0.155
1960-1969	Energy	0.099	Acousticness	-0.238
1970-1979	Energy	0.119	Acousticness	-0.223
1980-1989	Energy	0.113	Acousticness	-0.168
1990-1999	Explicit	0.134	Acousticness	-0.122
2000-2009	Loudness	0.118	Acousticness	-0.086
2010-2019	Explicit	0.155	Instrumentalness	-0.296
2020-2021	Explicit	0.23	Instrumentalness	-0.357

TABLE 1. Coefficient Table

K-Means classification was utilized to create nine clusters for all sixteen attributes, as shown below in Figure 7. In particular, Figure 7 depicts bar charts that illustrate the average for each of the nine clusters for all 16 attributes. From Figure 7, we calculated the correlation coefficients for each of the cluster's average for an attribute with each cluster's average popularity. Table 2 reproduced below shows that the correlation coefficients for attributes correlated with popularity ended up being highly positive or negative.

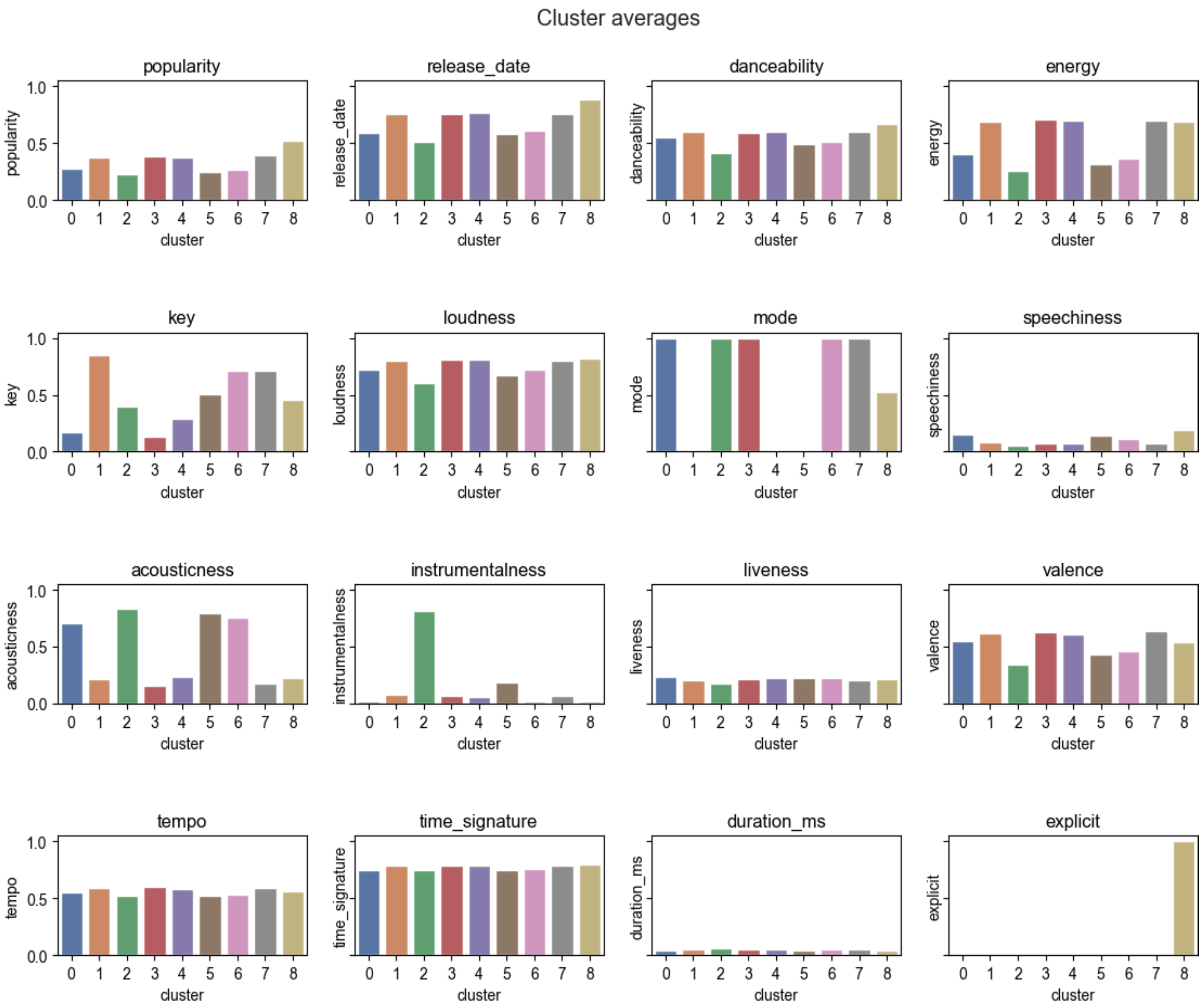


FIGURE 7. Averages for each of the nine clusters for all 16 attributes.

Attribute	Correlation Coefficient
Explicit and Popularity	0.740967
Release Date and Popularity	0.804947
Energy and Popularity	0.948962
Danceability and Popularity	0.732237
Loudness and Popularity	0.932758
Speechiness and Popularity	-0.759953
Instrumentalness and Popularity	-0.707906

Table 2: Cluster Correlation Coefficient Averages

APPLICATIONS:

Through several different predictive music models, we were able to identify the attributes, i.e., danceability, loudness, tempo, explicit, and energy, that are attributable to the popularity of a song or artist. With the use of the predictive music popularity models, the music industry can identify these attributes that contribute to overall popularity of a song or artist. Accordingly, with the help of the predictive music models, the music industry can compose songs that include attributes correlated with popularity. Similarly, the music industry in signing an artist to a record label can ensure that the artist has certain characteristics, e.g., high energy and danceability, that contribute to these attributes correlated to popularity. Thus, with the help of the predictive models, the music industry can harness data science to predict music popularity.

The predictive models can also be extended to music streaming platforms, such as Spotify, Pandora, and Apple Music, that can use the predictive powers of these models to create song recommendations based upon songs that contain similar attribute values, which have been predicted to contribute to overall popularity, and thus listenability. In other words, the models can be extended to recommend songs with similar attributes and values, e.g., danceability .70.

REFERENCES

[1] John Joyce: ‘Pandora and the music genome project: song structure analysis tools facilitate new music discovery,’ Scientific Computing, 2006, 23:43.

[2] Elbir, A., Aydin, N: ‘Music genre classification and music recommendation by using deep learning,’ Electronics Letters, 2020, 56:627-629.

[3] Tao, Li, Tzanetakis, G.: “factors in automatic musical genre classification of audio signals,” IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2003, 19-22.

[4] Holzapfel, A., and Stylianou, Y.: ‘Musical genre classification using non-negative matrix factorization-based features’, IEEE Trans. Audio Speech, Language Process, 2008, 16:424-434.

[5] ‘About the Music Genome Project’, Pandora Media, Inc. https://www.pandora.com/corporate/mgp.shtml

[6] ‘Web API Reference’, Spotify AB. https://developer.spotify.com/documentation/web-api/reference/#object-trackobject