School of Information Studies
Syracuse University

Syracuse University
School of Information Studies

IST 652: Predicting Song Popularity
Final Project Report

Project Introduction

The music industry has evolved a lot over the last several decades and will continue to evolve. This is largely in part due to advancements in technology. It is crazy to think that not too long ago CD players were still commonplace for listening to music, while nowadays, companies such as Spotify provide on demand streaming of music directly to any device. Not only has the method of listening to music changed due to technology, but the sound of the music itself has changed as well. For example, a lot of recordings are now digitally enhanced and new genres have emerged based on pure digital production such as electronic dance music and dubstep.

More recent developments in technology allow for attributes about songs to be extracted from an audio file. For example, the Organize Your Music tool, is an online tool that takes a selected song and gathers information about it. It gathers attributes such as the song tempo, energy, and danceability, among others. This enables a new perspective of music to be explored and analyzed, which leads to the purpose of this project.

The following project will explore a Kaggle dataset along with additional data gathered from Wikipedia, consisting of thousands of songs with certain attributes. These attributes are centered around the song popularity.

Having a popular song is vitally important, but it can mean many different things to an artist. One benefit it has is the influence and trajectory it can take an artists' career. One song can make or break a new artist. Many artists strive to go 'viral' with a song that can bring the spotlight that they are looking for. Producers of music and labels can also benefit from understanding what makes a song popular. Having that knowledge can help them create a hit song, and thus, improve their profitability and chances of breeding successful artists.

Analysis Questions

The following are analysis questions in exploring the datasets with the primary goal in mind of can a song's popularity be predicted based off certain song attributes.

1. What song genre has the most popular songs?

2. How have popular songs changed over time? Is there a particular period (time) where song popularity was higher?

3. How has song attributes change over time?

4. Are there certain song attributes/qualities that are correlated with popular songs?

5. Can a song's popularity be predicted from it's attributes?

Method of Analysis

The program created will follow the following methodology in answering the analysis questions:

1. Data Importation

2. Data Preparation

3. Data Analysis

4. Data Visualization

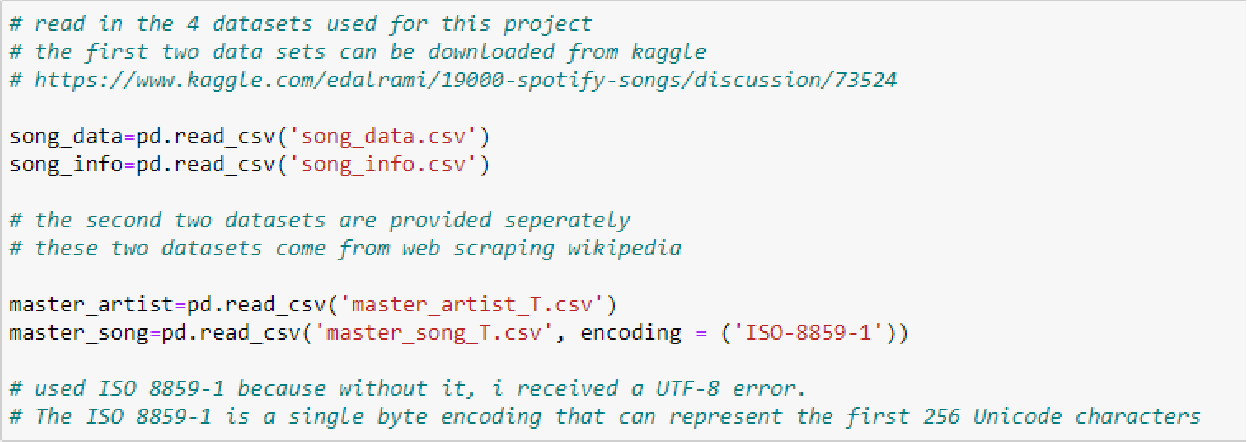
Data Importation and Data Preparation

Two original .CSV datasets from Kaggle, song_data and song_info, are used along with two additional datasets, master_song_T and master_artist_T, which were collected from unstructured data via web scraping wikepedia.

Web scraping was conducted to collect additional data about the songs and artists which was not included in the original datasets from Kaggle. For example, when a song was released is a crucial piece of information that was missing.

The code for the web scraping is not provided in the output of this report, since this was done using R studio vs Python. The data from the web scraping was packaged into two csv files for convenience use.

Each file was loaded in as a panda data frame for ease of use. Below is the meta data provided from http://organizeyourmusic.playlistmachinery.com/ *for the Kaggle datasets (song_data and song_info)*



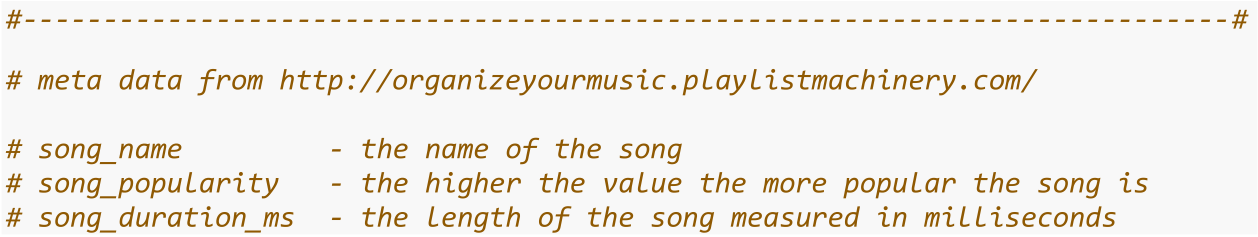
```
# read in the 4 datasets used for this project
# the first two data sets can be downloaded from kaggle
# https://www.kaggle.com/edalrami/19000-spotify-songs/discussion/73524

song_data=pd.read_csv('song_data.csv')
song_info=pd.read_csv('song_info.csv')

# the second two datasets are provided seperately
# these two datasets come from web scraping wikipedia

master_artist=pd.read_csv('master_artist_T.csv')
master_song=pd.read_csv('master_song_T.csv', encoding = ('ISO-8859-1'))

# used ISO 8859-1 because without it, i received a UTF-8 error.
# The ISO 8859-1 is a single byte encoding that can represent the first 256 Unicode characters
```

Dataframe: song_data

The song_data table is an original dataset from Kaggle, which consists of a collection of songs that were parsed via the Organize Your Music tool.

Note: there were many duplicate song names, without a way of uniquely identifying them. For this reason, duplicate song names were removed.

Original data had 18,835 row x 15 columns, using the drop.duplicate function, song_data reduce to 10,174 rows x 15 columns.



```
#----------------------------------------------------------------------------#

# meta data from http://organizeyourmusic.playlistmachinery.com/

# song_name          - the name of the song
# song_popularity    - the higher the value the more popular the song is
# song_duration_ms   - the length of the song measured in milliseconds
# acousticness       - the higher the value the more acoustic the song is
# danceability       - the higher the value, the easier it is to dance to
# energy             - the higher the value, the more energtic the song is
# instrumentalness   - the higher the value, the more instrumental the song is
# key                - description not provided
# liveness           - the higher the value, more likely a live recording
# loudness           - the higher the value, the louder, measured in dB
# audio_mode         - description not provided
# speechiness        - the higher the value the more spoken word in the song
# tempo              - the tempo of the song measured in beats per minute
# time_signature     - description not provided
# audio_valence      - the higher the value, the more positive mood

#---------------------------------------------------------------------------#
```

Dataframe: song_info

The song_info table is an original dataset from Kaggle which contains the corresponding artist, album, and playlist for each song in the song_data table.

Note: as in the song_data table, there were duplication errors in this table as well. This is resolved in the same manner by removing duplicate song names.

Original data had 18,835 row x 4 columns, using the drop.duplicate function, song_data reduce to 10,174 rows x 4 columns.

```
#---------------------------------------------------------------------------#

# song_name          - the name of the song
# artist_name        - the name of the corresponding artist(s)
# album_names        - the name of the corresponding album(s)
# playlist           - the name of the corresponding playlist(s)

#---------------------------------------------------------------------------#
```

Dataframe: master_song

The master_song table was collected by web scraping wikipedia pages for additional information about the songs.

Original data had 14,003 row x 9 columns, using the drop.duplicate function, master_song reduce to 10,174 rows x 10 columns.

```
#---------------------------------------------------------------------------#

# song_id            - identifier for the songs
# song_name          - the name of the song
# artist_name         -artist of the song
# song_single        - binary whether the song is a single or not
# song_released      - the year that the song was released in
# song_genre         - the corresponding genre(s) for the song
# song_label         - the corresponding label(s) for the song
# song_songwriter    - the corresponding songwriter(s) for the song
# song_producer      - the corresponding producer(s) for the song

#----------------------------------------------------------------------------#
```

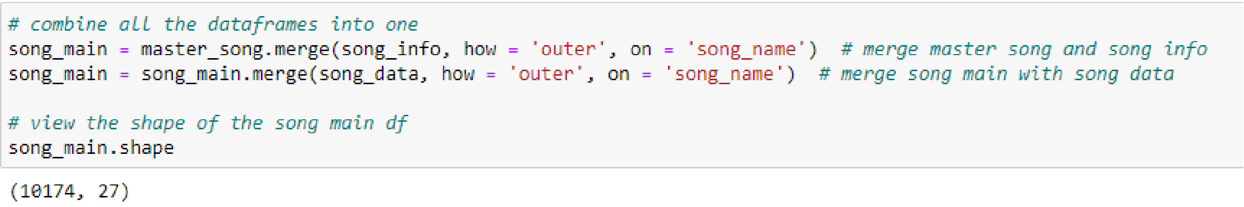
Dataframe: master_artist

The master_artist table was constructed by web scraping wikipedia pages for additional artist information.

It has 7,564 rows x 4 columns.

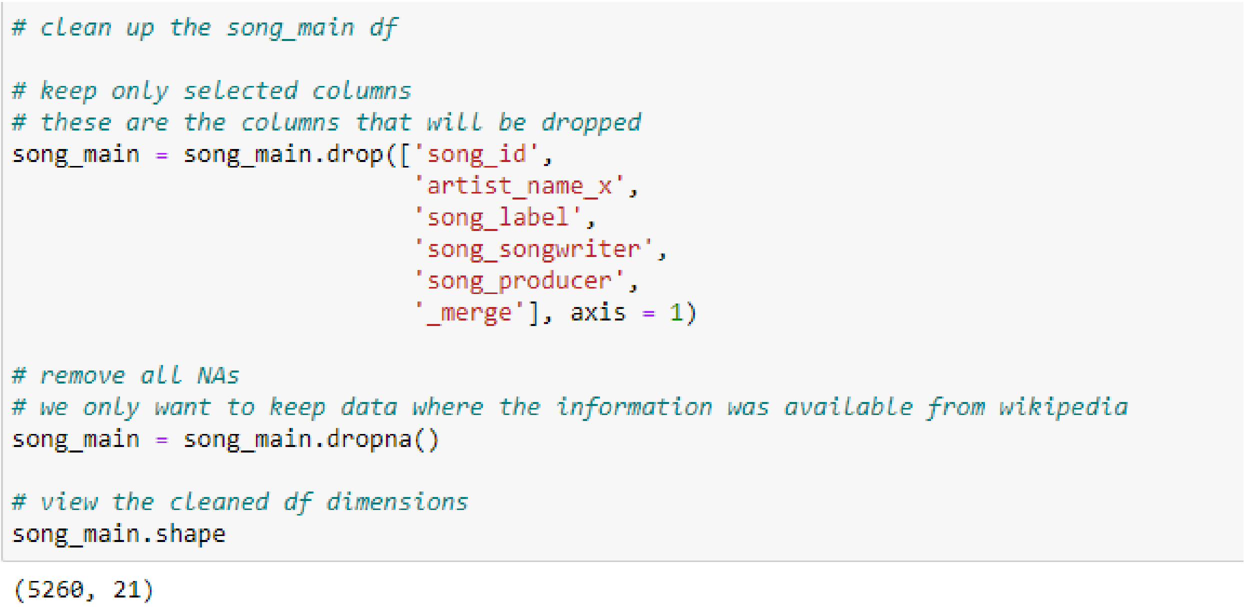
```
#----------------------------------------------------------------------------#

# artist_name        - the name of the artist
# birthday           - the date that the artist was born
# country            - 2 values - either USA or foreign
# startyear          - the year that the artist started making music

#----------------------------------------------------------------------------#
```

All dataframes were merged into one dataframe retitled as ‘song_main’ and then all NAs were removed with a final dataframe of 5,260 rows x 21 columns.



```
# combine all the dataframes into one
song_main = master_song.merge(song_info, how = 'outer', on = 'song_name')  # merge master song and song info
song_main = song_main.merge(song_data, how = 'outer', on = 'song_name')  # merge song main with song data

# view the shape of the song main df
song_main.shape
```

```
(10174, 27)
```



```
# clean up the song_main df

# keep only selected columns
# these are the columns that will be dropped
song_main = song_main.drop(['song_id',
                            'artist_name_x',
                            'song_label',
                            'song_songwriter',
                            'song_producer',
                            '_merge'], axis = 1)

# remove all NAs
# we only want to keep data where the information was available from wikipedia
song_main = song_main.dropna()

# view the cleaned df dimensions
song_main.shape
```

```
(5260, 21)
```

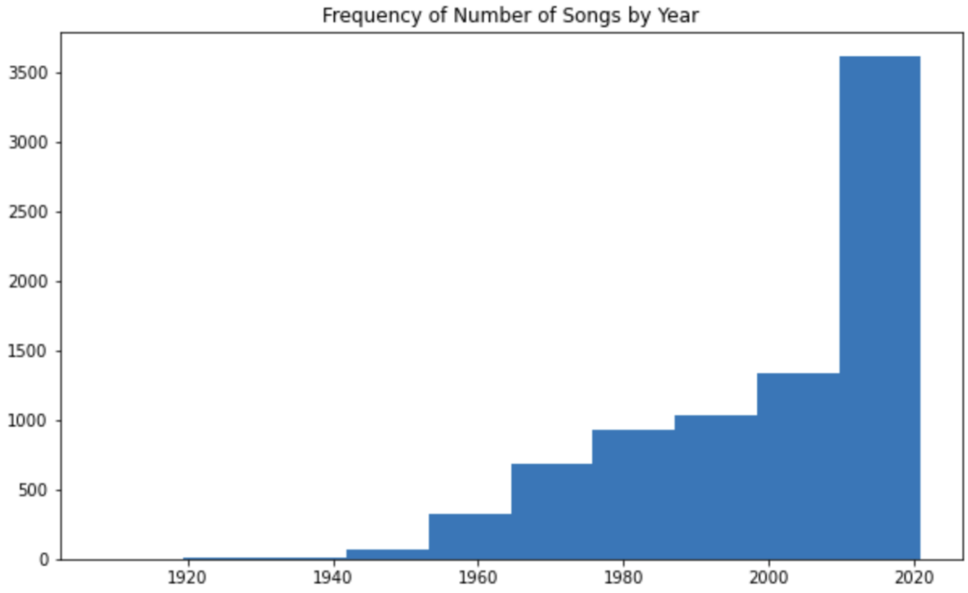
Data Analysis and Data Visualization

Initial Data Exploration

Song Release Year

The song release year was one of the attributes that was retrieved from web scraping Wikipedia. Here is how the distribution of the song release years looks.

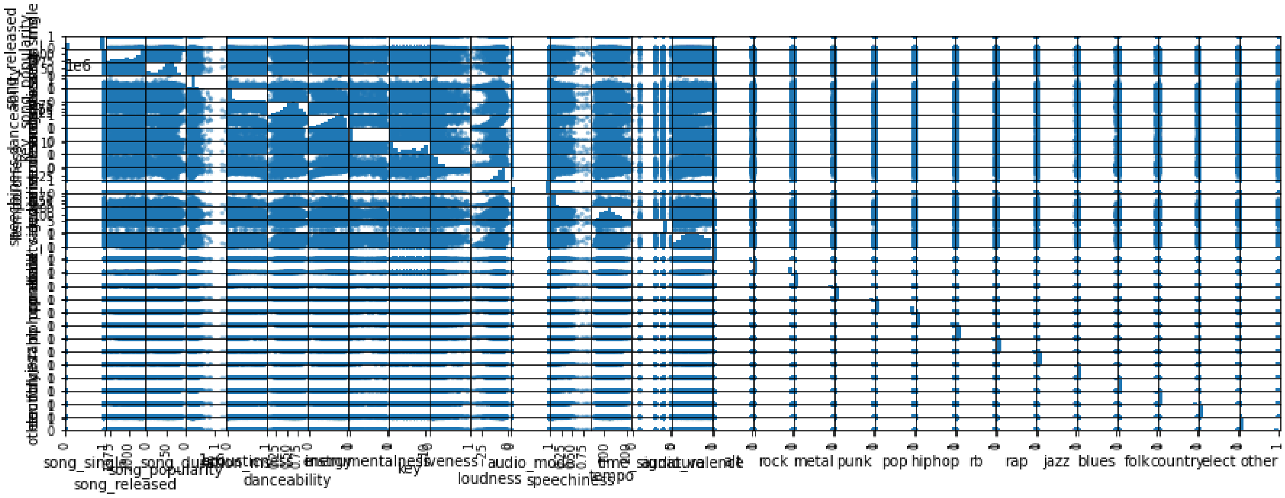
Note: there are a large number of missing song years due to the data not being available on Wikipedia which are not displayed on the graph below.



Observations
- there are more newer generation songs than older generation songs
- there is a particularly larger number of songs between 2010 and 2020
- there is a large spike in the number of songs released in 2017-2018
- the bias is probably because the creator of the data prefers newer songs

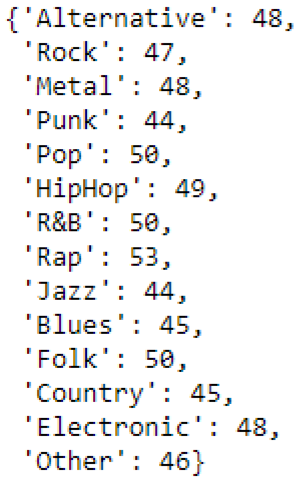
Additionally, we tested out pandas' scatter plot matrix to see if at a quick glance there's a visual relationship between combinations of variables. The matrix of scatter plots is used to visualize bivariate relationships between combinations of variables, each scatter plot in the matrix visualizes the relationship between a pair of variables, allowing many relationships to be explored in one chart.

However, upon looking at the matrix, there are too many variables compact together to quickly see a relationship.



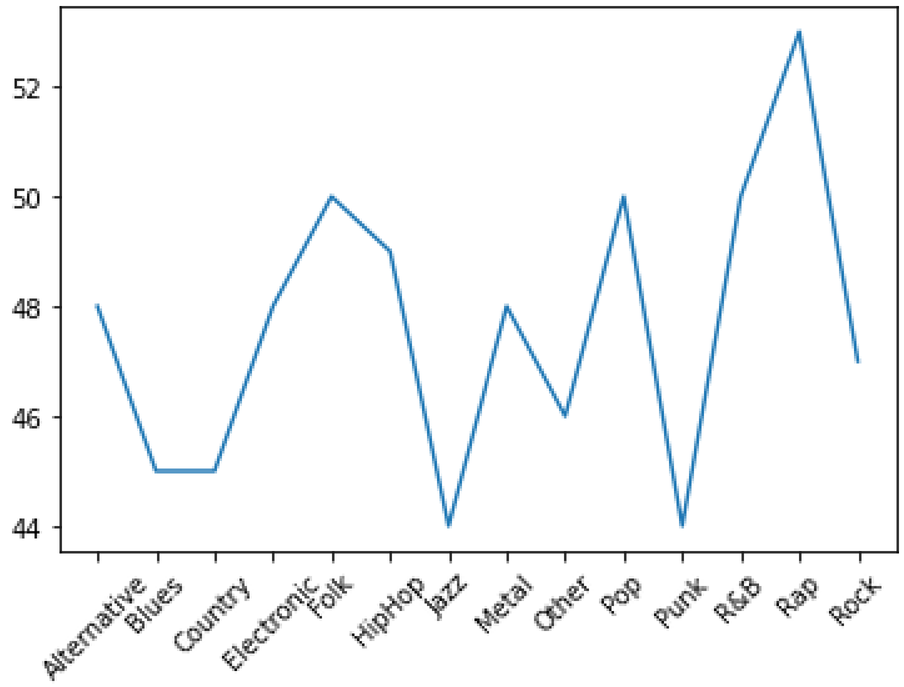
Question 1: What song genre has the most popular songs?

We unpacked the song genre within the data frame by creating a function which searches for a substring. There were 14 genres, of which multiple songs could fit within multiple genres. The function iterated through each of the songs and corresponding genres and if it contained the genre, it will provide a 1 for true or a 0 for false. Below is the result:



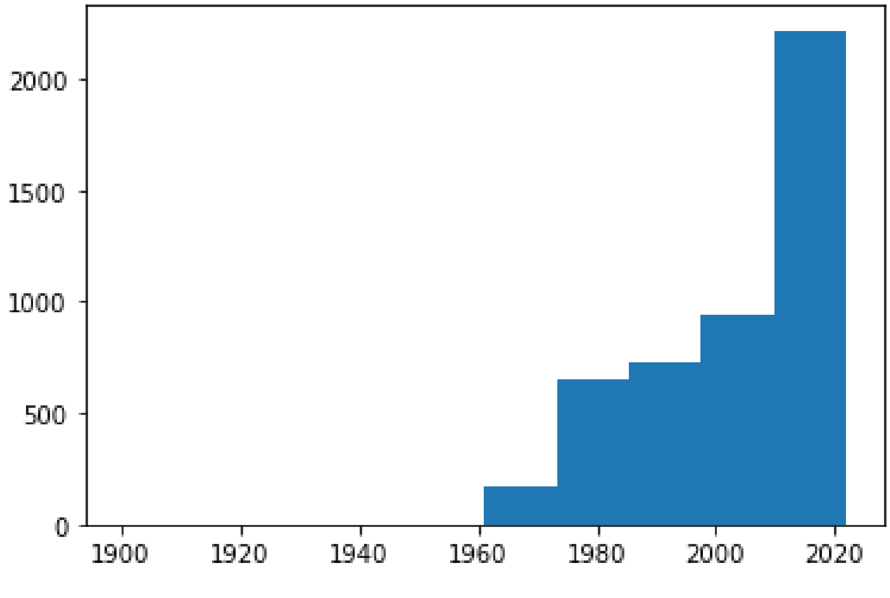
```
{'Alternative': 48,
 'Rock': 47,
 'Metal': 48,
 'Punk': 44,
 'Pop': 50,
 'HipHop': 49,
 'R&B': 50,
 'Rap': 53,
 'Jazz': 44,
 'Blues': 45,
 'Folk': 50,
 'Country': 45,
 'Electronic': 48,
 'Other': 46}
```

Observation: Rap genre has the most popular songs identified in this genre

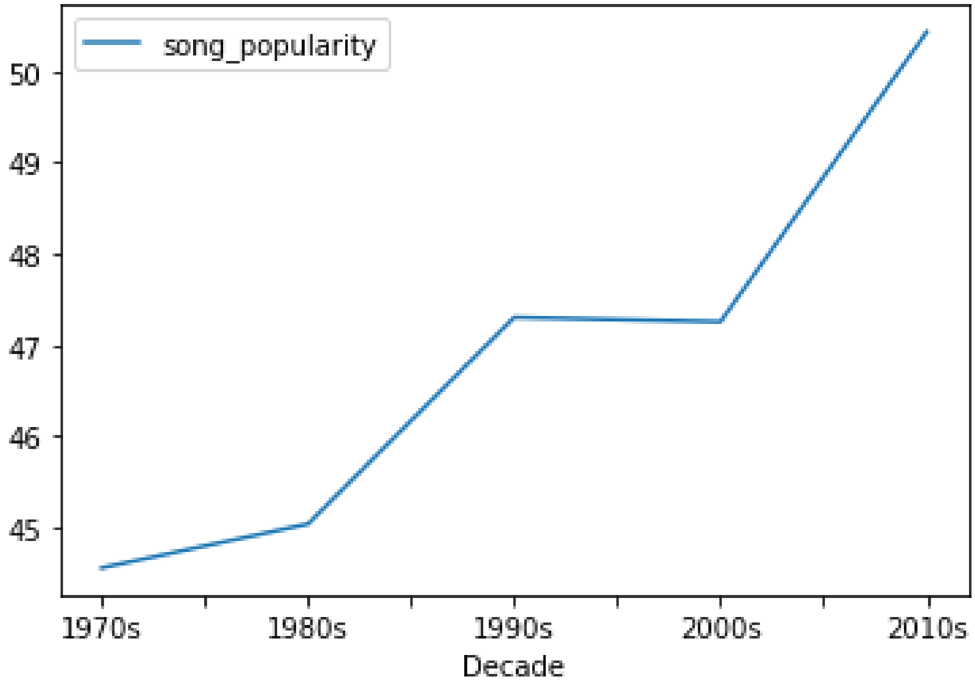


Question 2: How have popular songs changed over time? Is there a particular period (time) where song popularity was higher?

To view how popular songs have changed over time, and which particular period of time had the highest song popularity, we looked at song released data from 1969 – 2021. A histogram shown below indicates that 2000s had a high level of popular songs.

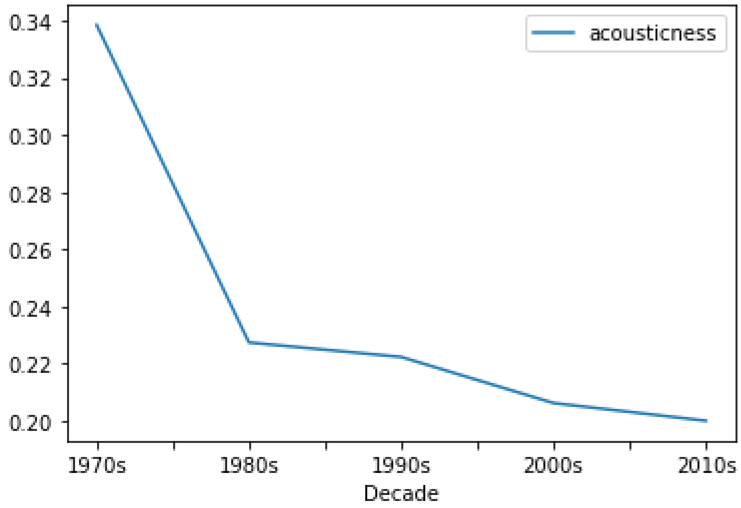


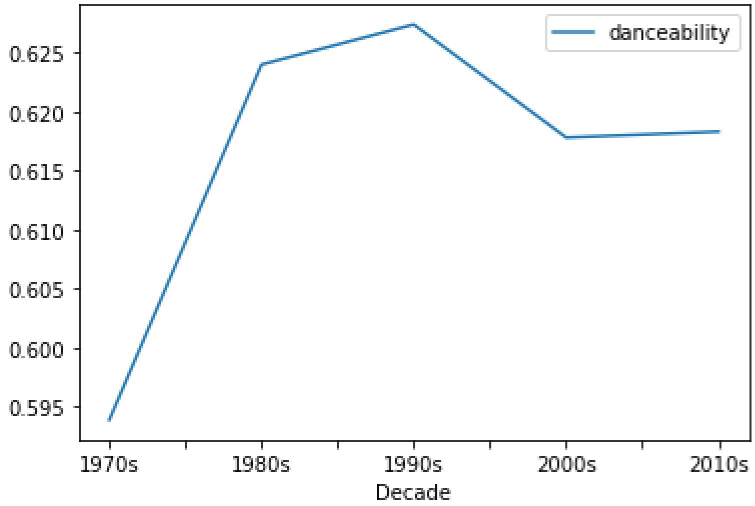
We also bin the released year by decades to demonstrate this visual. From the below line chart, it is clear to see that the average song popularity was steadily increasing between the 1970s - 1990s before plateauing between the 1990 and 2000. Since then, it has been on the rise, increasing to 50.54% in the 2010s.

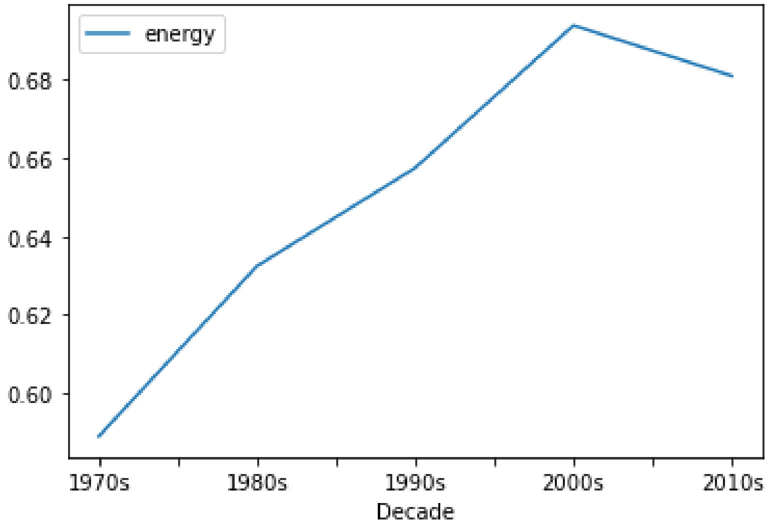


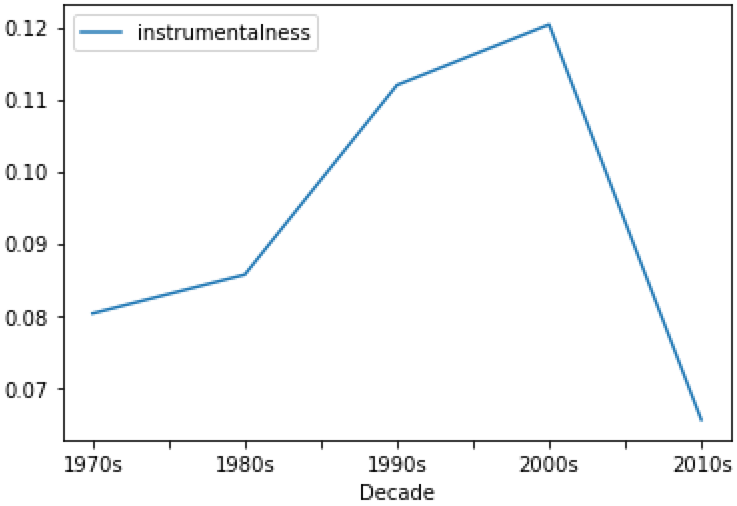
Question 3: How have song attributes changed over years?

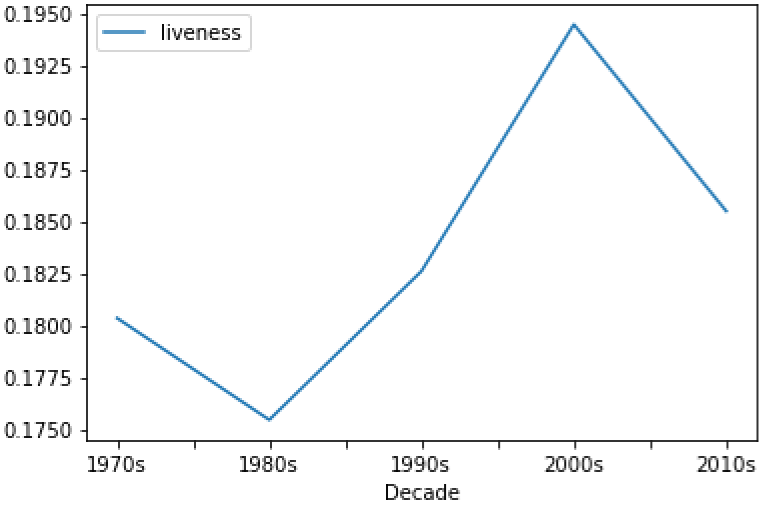
Since we bin the decades per question 2, we wanted to look at how each song attribute has changed over the years, below are the visuals of how each attributes performed:

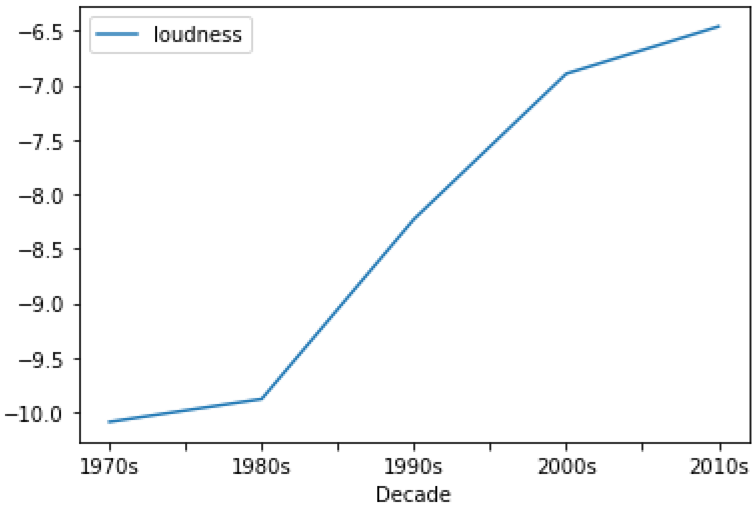

0.34
0.32
0.30
0.28
0.26
0.24
0.22
0.20
acousticness
1970s
1980s
1990s
2000s
2010s
Decade

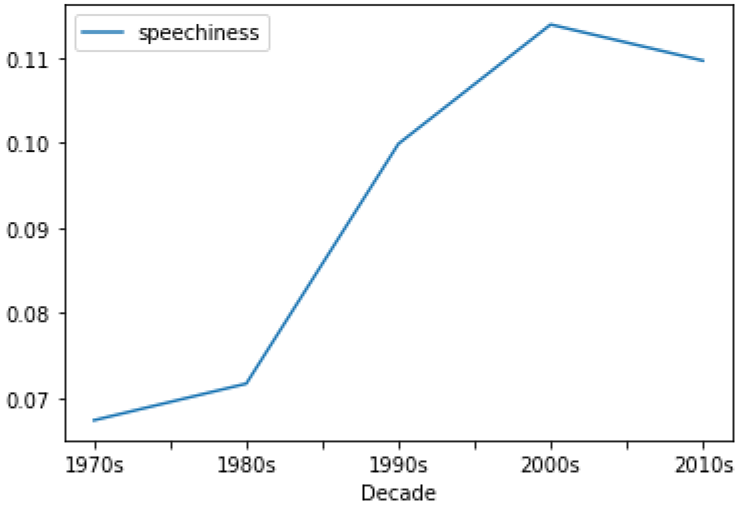

0.625
0.620
0.615
0.610
0.605
0.600
0.595
danceability
1970s
1980s
1990s
2000s
2010s
Decade

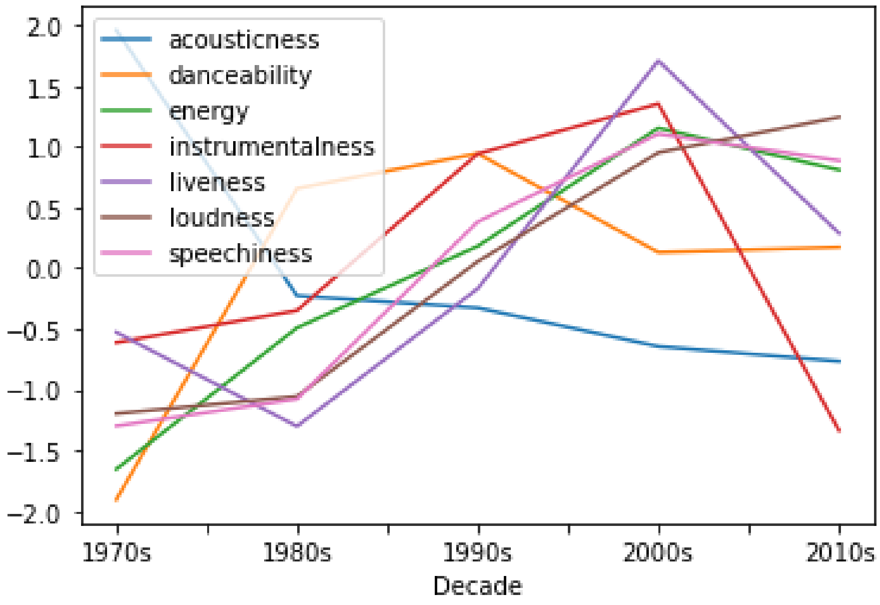

0.68
0.66
0.64
0.62
0.60
energy
1970s
1980s
1990s
2000s
2010s
Decade


instrumentalness
0.12
0.11
0.10
0.09
0.08
0.07
1970s
1980s
1990s
2000s
2010s
Decade


liveness
0.1950
0.1925
0.1900
0.1875
0.1850
0.1825
0.1800
0.1775
0.1750
1970s
1980s
1990s
2000s
2010s
Decade


loudness
−6.5
−7.0
−7.5
−8.0
−8.5
−9.0
−9.5
−10.0
1970s
1980s
1990s
2000s
2010s
Decade

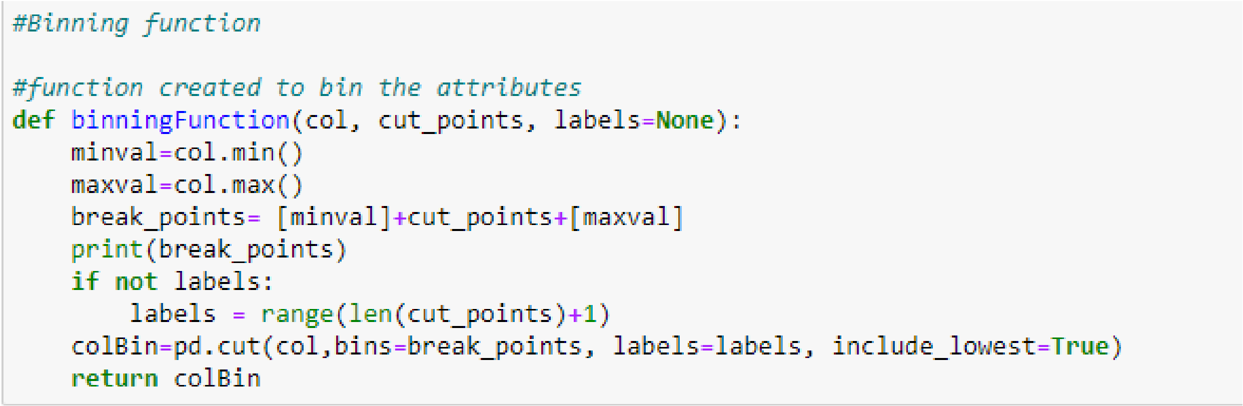

speechiness
0.11
0.10
0.09
0.08
0.07
1970s
1980s
1990s
2000s
2010s
Decade


acousticness
danceability
energy
instrumentalness
liveness
loudness
speechiness
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
1970s
1980s
1990s
2000s
2010s
Decade

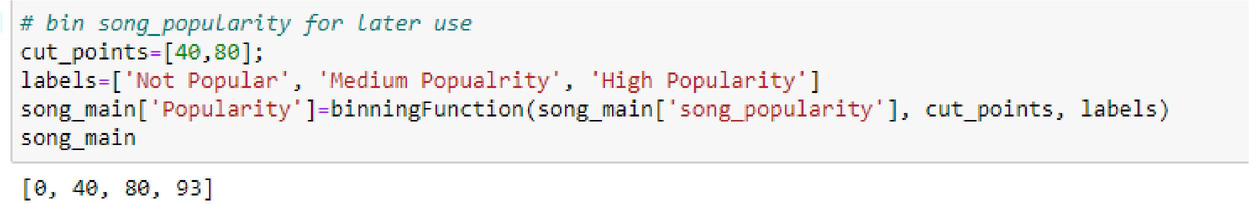
Acousticness is the one attribute that has been on the decline throughout the decades. This is likely the result of electronic instruments becoming more prominent. Instrumentalness, Liveness, and Speechiness all increased over time before declining in the 2010s. Energy and Loudness have been heavily increasing over the years.

Question 4: Are there certain song attribute that correlated with popular songs?

We binned the popular songs from the continuous variable to a discrete variable of ‘Not Popular’, ‘Medium Popularity’, and ‘High Popularity’ through a binning function created:



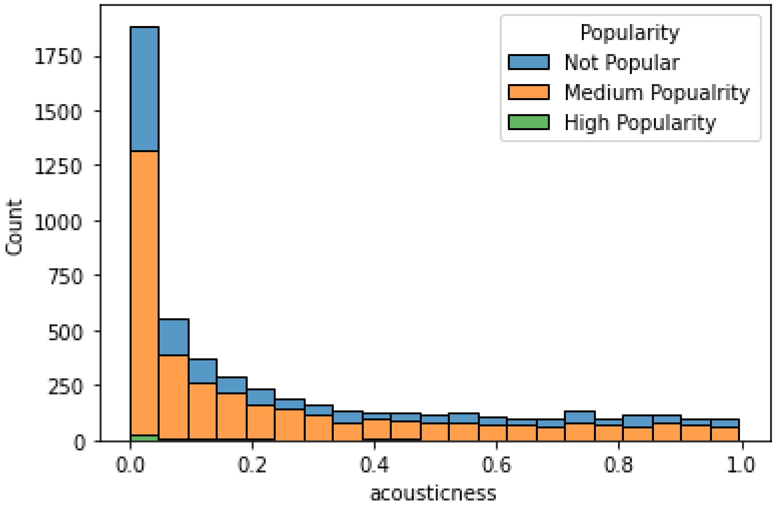
```
#Binning function

#function created to bin the attributes
def binningFunction(col, cut_points, labels=None):
    minval=col.min()
    maxval=col.max()
    break_points= [minval]+cut_points+[maxval]
    print(break_points)
    if not labels:
        labels = range(len(cut_points)+1)
    colBin=pd.cut(col,bins=break_points, labels=labels, include_lowest=True)
    return colBin
```

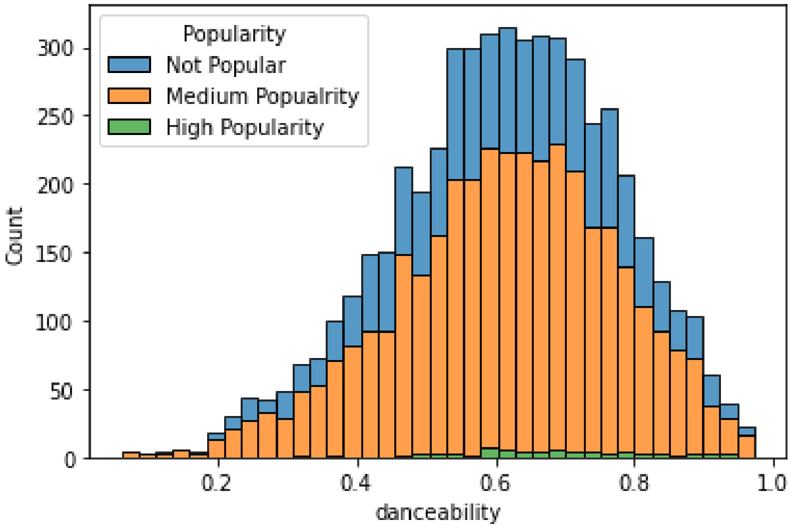


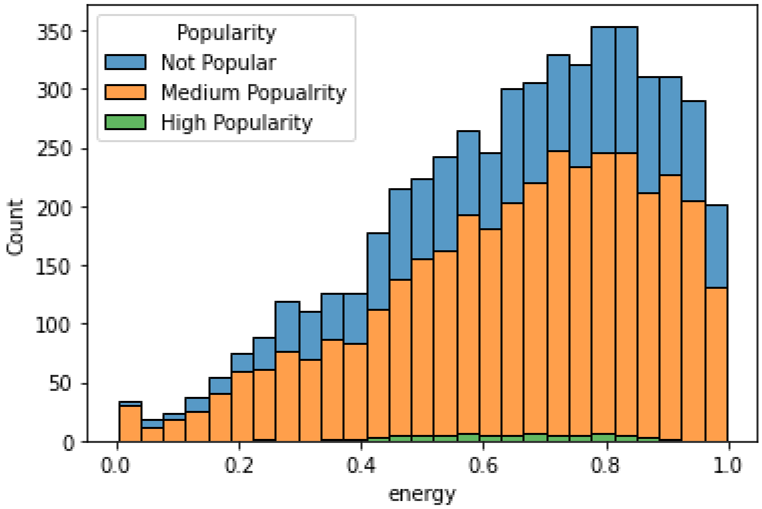
```
# bin song_popularity for later use
cut_points=[40,80];
labels=['Not Popular', 'Medium Popualrity', 'High Popularity']
song_main['Popularity']=binningFunction(song_main['song_popularity'], cut_points, labels)
song_main
```

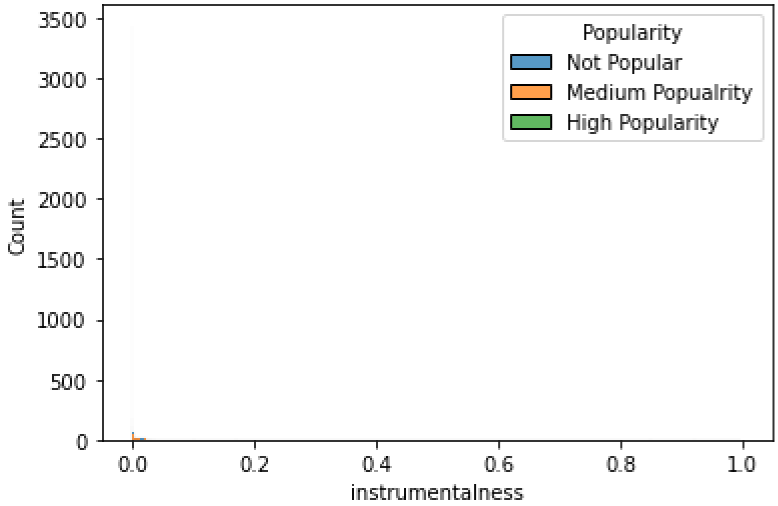
```
[0, 40, 80, 93]
```

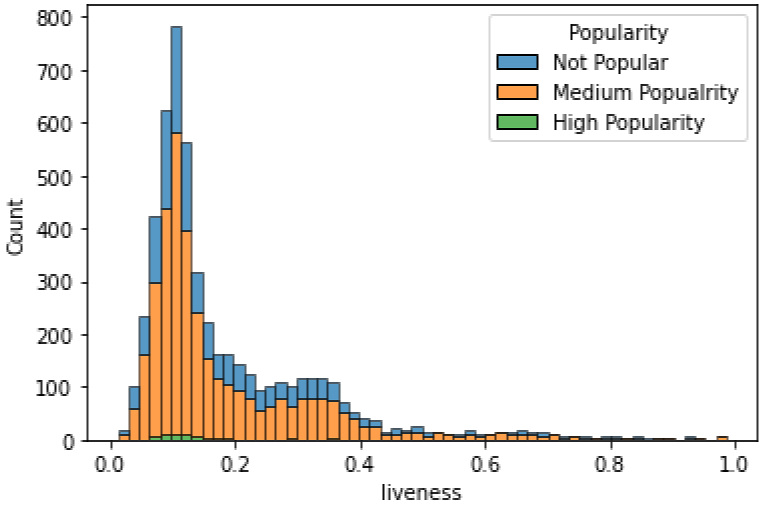
We used the seaborn package based on matplotlib to visually showed the relationship interface between each attribute and song popularity. Below are the results:

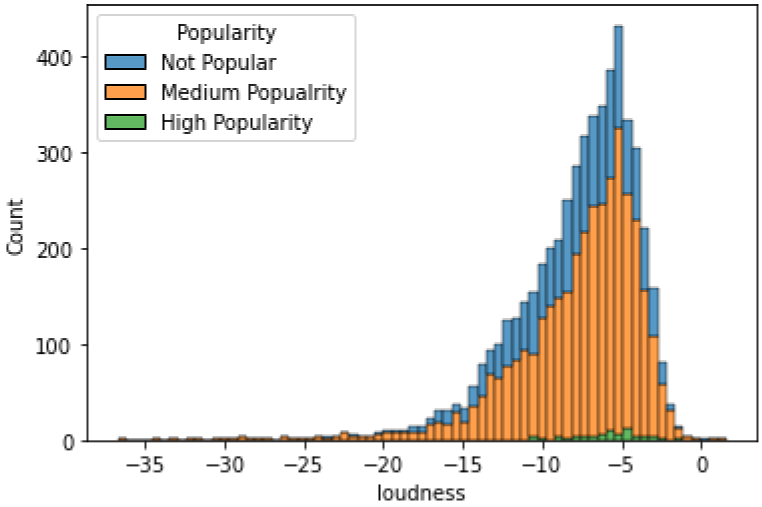

Popularity
🟦 Not Popular
🟧 Medium Popualrity
🟩 High Popularity
Count
0
250
500
750
1000
1250
1500
1750
0.0
0.2
0.4
0.6
0.8
1.0
acousticness

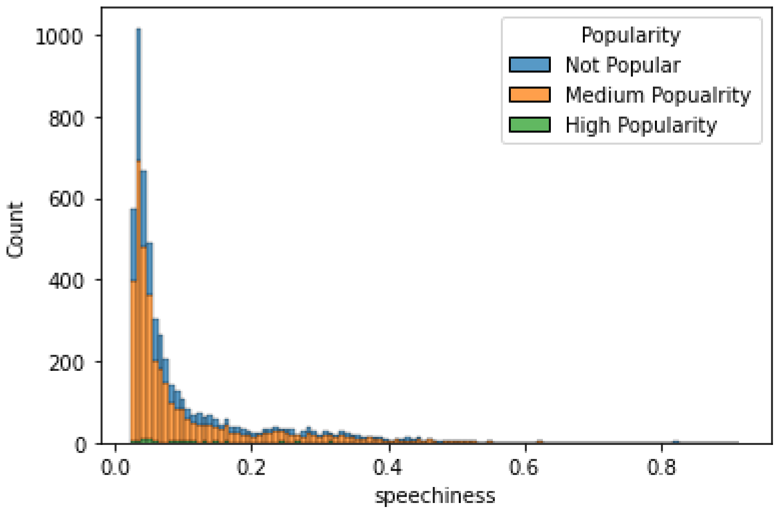

Popularity
🟦 Not Popular
🟧 Medium Popualrity
🟩 High Popularity
Count
0
50
100
150
200
250
300
0.2
0.4
0.6
0.8
1.0
danceability

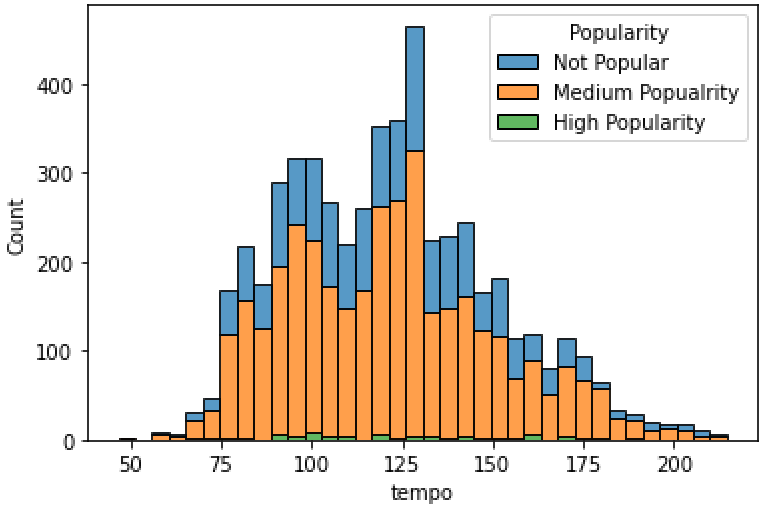

Popularity
🟦 Not Popular
🟧 Medium Popualrity
🟩 High Popularity
Count
0
50
100
150
200
250
300
350
0.0
0.2
0.4
0.6
0.8
1.0
energy

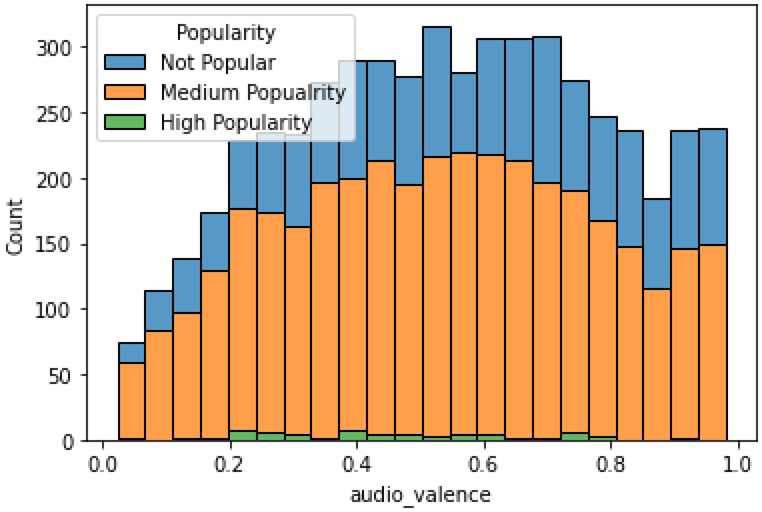

Popularity
Not Popular
Medium Popualrity
High Popularity
Count
3500
3000
2500
2000
1500
1000
500
0
0.0
0.2
0.4
0.6
0.8
1.0
instrumentalness


Popularity
Not Popular
Medium Popualrity
High Popularity
Count
800
700
600
500
400
300
200
100
0
0.0
0.2
0.4
0.6
0.8
1.0
liveness


Popularity
Not Popular
Medium Popualrity
High Popularity
Count
400
300
200
100
0
−35
−30
−25
−20
−15
−10
−5
0
loudness


Popularity
Not Popular
Medium Popualrity
High Popularity
Count
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
speechiness


Popularity
Not Popular
Medium Popualrity
High Popularity
Count
400
300
200
100
0
50
75
100
125
150
175
200
tempo


Popularity
Not Popular
Medium Popualrity
High Popularity
Count
300
250
200
150
100
50
0
0.0
0.2
0.4
0.6
0.8
1.0
audio_valence

Attributes of a popular songs seems to have the following attributes: low acousticness, high danceability, high energy, low instrument, not a live recording(liveness), high loudness, low speech, and medium tempo. The audio valence measuring the positivity mood of the song seem not to have a definitive conclusion, there's popularity songs across the x axis.

Group Member Tasks & Roles
A breakdown of roles and responsibilities is found below. However, most components were developed in close collaboration with our teammates.

- Data collection — Tyler, Annie, Dan
- Web scrapping — Tyler
- Data cleanup, preprocessing and preparation- Tyler, Annie, Dan
- Question 1analysis- Tyler
- Question 2 and 3 analysis - Dan
- Question 4 analysis - Annie
- Paper Write-up — Tyler, Annie, Dan
- Presentation – Tyler, Annie, Dan

Project Conclusion

The goal of this project and this analysis is to answer if a song's popularity can be determined from its attributes. Through the analysis, it is easy to see that popular song do have common attributes such as low acousticness, high danceability, high energy, medium tempo and so forth. However, song popularity has changed drastically over the decades, due to culture and generation interest at the time of the song. We have seen that this can change a song's popularity.

Being able to predict the popularity of a song, as mentioned earlier is important to the artist to help them 'make it' in the music industry but also to music producers and music labels to profit from the songs/artist and continue breeding successful artists. Additionally, it is important to the consumers as well to see songs become popular especially if it's coming from their favorite artist(s). This analysis has taught that song attributes are important to a song's popularity but social interest/disinterest/culture/etc. are necessary additional information needed to the making of a successful song.

Final thoughts and next steps

The results from this project were promising. This project shows that it is possible to predict the popularity of a song. If more resources were invested to improve the quality of the data, or to collect more data such as additional song attributes, and also relevant events/social changes happening during the song's release year, this could improve the accuracy of predicting the song popularity. This project only scratches the surface of what can potentially be done in the music industry, but it does provide a foundation for analytics in music that can continue to built on.