Spotify Data Visualization and Analysis

by

SAKTHI BALA

Table Of Contents

In a Spotify music analysis project, various libraries and approaches are typically used to analyze and gain insights from music-related data. Below are the commonly used libraries and a high-level approach to solving problems in this analysis:

Why This Dataset?

As most of the people like to listen music, would be interesting to analyse the one of the most popular fields in the world. Ultimately, working with music datasets is enjoyable and often feels more like a hobby than work, making it a fulfilling pursuit for music lovers. We can analyse this dataset to find out the popular artists, popular songs, People’s taste in music, etc.

Music datasets encompass a wide variety of genres, languages, and cultures, offering the opportunity to explore music diversity and global trends. We can use this dataset for various actions like building a music recommendation system, Hit song predictions, Music genre analysis, Comparisons of Musicians, Sentiment analysis, etc

DOMAIN

The music industry is a multifaceted and ever-evolving ecosystem that has undergone significant transformations in recent decades. As we embark on an analysis of the music industry, we aim to explore its complexities, challenges, and opportunities while leveraging data driven insights to understand its current landscape.

1. Industry Significance:

The music industry holds a vital place in global culture and entertainment, acting as a powerful medium for artistic expression and emotional connection Its economic impact is substantial, contributing to both the creative and business sectors of the global economy.

2. Industry Segments:

The music industry comprises several key segments, including music creation and production, distribution, live events and concerts, music streaming, and music publishing. Each segment plays a distinct role in the creation, promotion, and distribution of music content.

3. Digital Transformation:

The industry has experienced a profound digital transformation, with the advent of digital downloads, streaming platforms, and online distribution. The shift from physical to digital formats has had far-reaching implications for artists, labels, and consumers.

4. The Role of Streaming:

Streaming platforms, such as Spotify, Apple Music, and Amazon Music, have become dominant forces in music distribution. They offer new revenue models, impact artist discovery, and shape consumer preferences.

5. Market Dynamics:

The music industry is highly competitive, with numerous players vying for market share. The emergence of independent artists, streaming exclusivity deals, and live event dynamics add complexity to the market.

6. Revenue Models:

Explore the various revenue models within the industry, including music sales, licensing, advertising, and subscription-based streaming. Understand the impact of these models on artists, labels, and consumers.

7. Artist and Fan Engagement:

Discuss the changing relationship between artists and their fan base in the digital age. Highlight the role of social media, crowdfunding, and direct-to-fan engagement.

8. Challenges and Opportunities:

Identify challenges such as copyright issues, piracy, and the evolving role of record labels. Explore opportunities in emerging markets, innovative technologies, and new revenue streams.

Libraries Used:

Pandas: Pandas is used for data manipulation, cleaning, and preparation. It allows you to work with tabular data efficiently.

Matplotlib and Seaborn: These libraries are used for data visualization, creating charts, graphs, and plots to represent the analysis results.

Numpy: NumPy provides support for mathematical operations and arrays, making it useful for numerical data analysis.

Approach to Solve Problems in Spotify Music Analysis:

Data Collection: Gather the dataset containing music-related data, including attributes like tempo, loudness, danceability, genre, artist information, and popularity (if available).

Data Preprocessing:

Handle missing data: Check for missing values and decide how to handle them, either by imputation or removal.

Data cleaning: Remove duplicate entries and perform any necessary data cleaning.

Feature engineering: Create new features or transform existing ones to extract meaningful insights.

Exploratory Data Analysis (EDA):

Visualize data: Use Matplotlib and Seaborn to create visualizations such as histograms, scatter plots, and box plots to understand the distribution and relationships between variables.

Summary statistics: Calculate and analyze summary statistics for key attributes.

Outlier Detection and Handling:

Identify outliers using statistical methods like IQR (Interquartile Range) and Z-Score.

Decide whether to remove or transform outliers based on domain knowledge and analysis goals.

Genre Analysis:

Analyze the distribution of music genres.

Explore genre preferences among users.

Visualize how genres have evolved over time.

Artist Analysis:

Explore the characteristics of different artists' catalogs.

Analyze how artist attributes impact song popularity.

Temporal Trends:

Investigate time-based trends or patterns in music attributes.

Analyze how audio features have changed over the years.

Correlation Analysis:

Calculate correlations between variables to identify relationships.

Visualize correlations using heatmaps.

Popularity Analysis:

Determine the factors associated with song popularity.

Group and analyze songs by genre to find the most popular genres.

Visualization and Reporting:

Create comprehensive visualizations and reports to communicate findings and insights effectively.

Use storytelling techniques to present results clearly and engage the audience.

Conclusion and Insights:

Summarize the key findings and insights from the analysis.

Provide recommendations or conclusions based on the analysis results.

Data Understanding

The Data contains 1000 rows and 17 columns.

The columns it contains are as follows.

- **Song_title**: The name of the title.

- **Artist:** The singer or the group of singers who produced the song.

- **Year Released:** The Year at which the song was released.

- **Acousticness:** A measure of the extent to which a musical track relies on acoustic instruments and natural sounds as opposed to electronic or synthesized elements.

- **Danceability:** A numerical audio feature that assesses the suitability of a music track for dancing, typically based on rhythm, tempo, and beat characteristics, with higher values indicating tracks that are more dance friendly.

- **Duration_ms:** The duration of the music in milli-seconds.

- **Energy:** A numerical audio feature that quantifies the intensity and activity level of a musical track, with higher values indicating more energetic and dynamic music and lower values indicating calmer or less intense compositions.

- **Instrumentalness:** A numerical audio feature that measures the likelihood of a song being purely instrumental (without vocals), with higher values indicating a higher probability of instrumental music and lower values indicating the presence of vocals or lyrics in the track.

- **Key:** A musical attribute that indicates the tonic or central note upon which a piece of music is based, defining the overall pitch and tonal centre of a composition.

- **Liveness:** An audio feature that assesses the likelihood of a musical performance being live, with higher values indicating a greater probability of a live performance and lower values suggesting a studio recording or electronically produced track.

• **Loudness:** A measure of the overall volume or amplitude of a musical track, often expressed in decibels (dB), indicating how loud or quiet a piece of music sounds when played.

• **Mode:** A musical attribute that identifies whether a piece of music is predominantly in a major key (associated with a brighter and happier sound) or a minor key (associated with a darker and sadder sound), which helps define the emotional character of the composition.

• **Speechiness:** A numerical audio feature that quantifies the presence of spoken words or speech-like elements in a musical track, with higher values indicating a greater proportion of spoken content, making it suitable for speech or spoken word rather than singing or instrumental music.

• **Tempo:** A numerical audio feature that represents the speed or pace of a musical composition, typically measured in beats per minute (BPM), indicating how fast or slow the rhythm or pulse of the music is.

• **Time_signature:** A musical notation indicating the number of beats per measure and the type of note that receives one beat, defining a piece's rhythmic structure. For example, 4/4 denotes four beats per measure with a quarter note as one beat.

• **Valence:** A numerical audio feature representing the emotional positivity or negativity of a musical track, with higher values indicating a more positive, happy, or cheerful mood, and lower values suggesting a more negative, sad, or melancholic mood.

• **Target:** The term "target" typically refers to a numerical value or score associated with a specific attribute or characteristic of a musical track.

• **Genre:** "Genre" refers to a categorization or classification of songs based on shared musical characteristics, styles, themes, or cultural influences, allowing listeners to identify and describe different types or styles of music.

• **Artist Type:** It is the numbers of artists like solo, duo or group.

Questions for Analysis

1. What are the names and data types of the columns?

2. What are the basic summary statistics?

3. Are there any categorical variables and missing values? If so, print it.

4. Are there any outliers in the data? If so, use box plots, histograms and visualize.

5. Is the data balanced or imbalanced? Visualize.

6. What is the target variable (if any).

7. What are the units of measurement for numerical columns? (Example: time, currency, date, distance)

8. Do you have domain clarification? Brief it.

9. Are there any time-based trends or patterns?

10. Are there any correlations between variables? Calculate correlations.

11. Which music genres are the most popular among users?

12. What features (e.g., acousticness, danceability) are associated with popular songs?

13. How do different artists' catalogues compare in terms of acoustic features?

14. Are there trends in audio feature preferences across different genres?

15. Explore which audio attributes, such as danceability, energy, and valence, are commonly associated with top-charting songs.

16. How have music genre preferences evolved over time?

17. Can we identify common patterns in songs with specific audio features?

18. Are there notable changes in audio features like tempo, loudness, or Instrumentalness in songs over time?

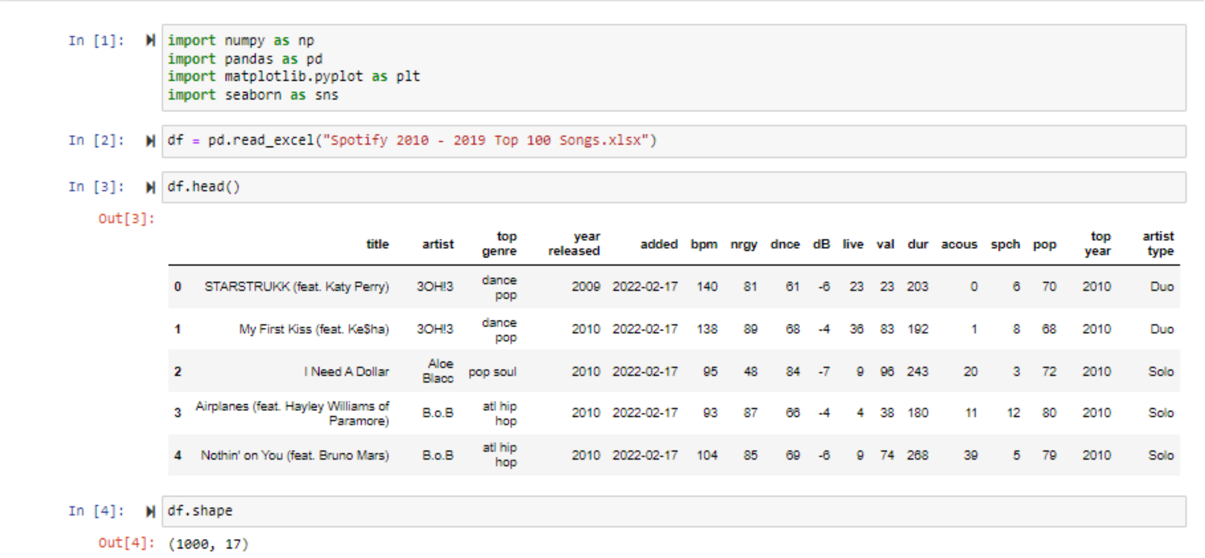
19. Is there relationship between the top and artist type?

20. Are there tempo preferences associated with specific music genres, and can we determine the typical tempos for different genres?

Answers to the Questions

I have collected the data set from Kaggle open-source platform and imported it.

First, we need to import the necessary packages and data and viewing the first 5 rows using the head () function.

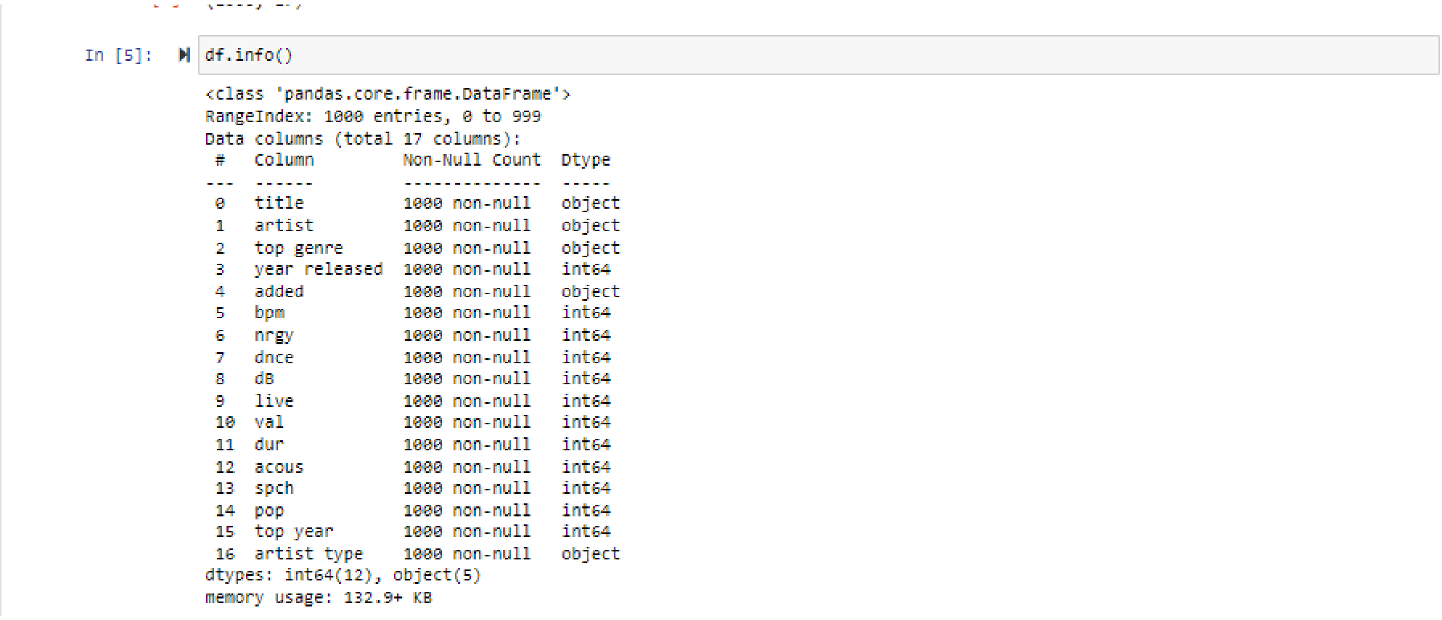


```
In [1]:  import numpy as np
          import pandas as pd
          import matplotlib.pyplot as plt
          import seaborn as sns

In [2]:  df = pd.read_excel("Spotify 2010 - 2019 Top 100 Songs.xlsx")

In [3]:  df.head()
Out[3]:
```

	title	artist	top genre	year released	added	bpm	nrgy	dnce	dB	live	val	dur	acous	spch	pop	top year	artist type
0	STARSTRUKK (feat. Katy Perry)	3OH!3	dance pop	2009	2022-02-17	140	81	61	-6	23	23	203	0	6	70	2010	Duo
1	My First Kiss (feat. Ke$ha)	3OH!3	dance pop	2010	2022-02-17	138	89	68	-4	36	83	192	1	8	68	2010	Duo
2	I Need A Dollar	Aloe Blacc	pop soul	2010	2022-02-17	95	48	84	-7	9	96	243	20	3	72	2010	Solo
3	Airplanes (feat. Hayley Williams of Paramore)	B.o.B	atl hip hop	2010	2022-02-17	93	87	66	-4	4	38	180	11	12	80	2010	Solo
4	Nothin' on You (feat. Bruno Mars)	B.o.B	atl hip hop	2010	2022-02-17	104	85	69	-6	9	74	268	39	5	79	2010	Solo

```
In [4]:  df.shape
Out[4]:  (1000, 17)
```

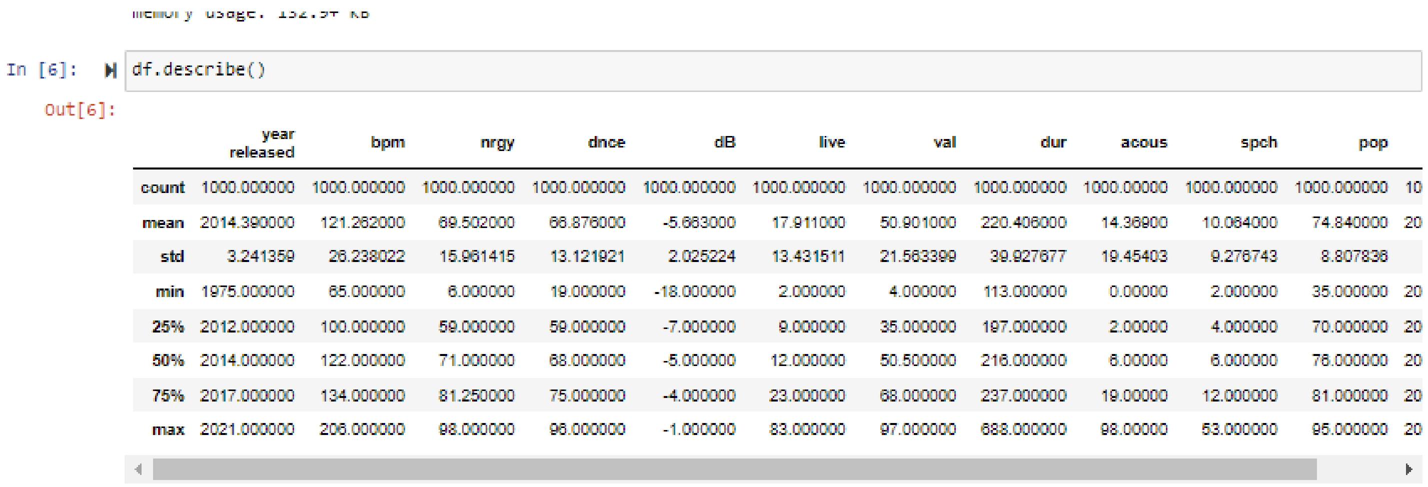
1. What are the names and data types of the columns?



```
In [5]:  df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 17 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   title          1000 non-null   object
 1   artist         1000 non-null   object
 2   top genre      1000 non-null   object
 3   year released  1000 non-null   int64 
 4   added          1000 non-null   object
 5   bpm            1000 non-null   int64 
 6   nrgy           1000 non-null   int64 
 7   dnce           1000 non-null   int64 
 8   dB             1000 non-null   int64 
 9   live           1000 non-null   int64 
 10  val            1000 non-null   int64 
 11  dur            1000 non-null   int64 
 12  acous          1000 non-null   int64 
 13  spch           1000 non-null   int64 
 14  pop            1000 non-null   int64 
 15  top year       1000 non-null   int64 
 16  artist type    1000 non-null   object
dtypes: int64(12), object(5)
memory usage: 132.9+ KB
```

I have used info() function to view the column along with the datatype of the columns. The Dtype indicates the data types of each column.

2. What are the basic summary statistics?

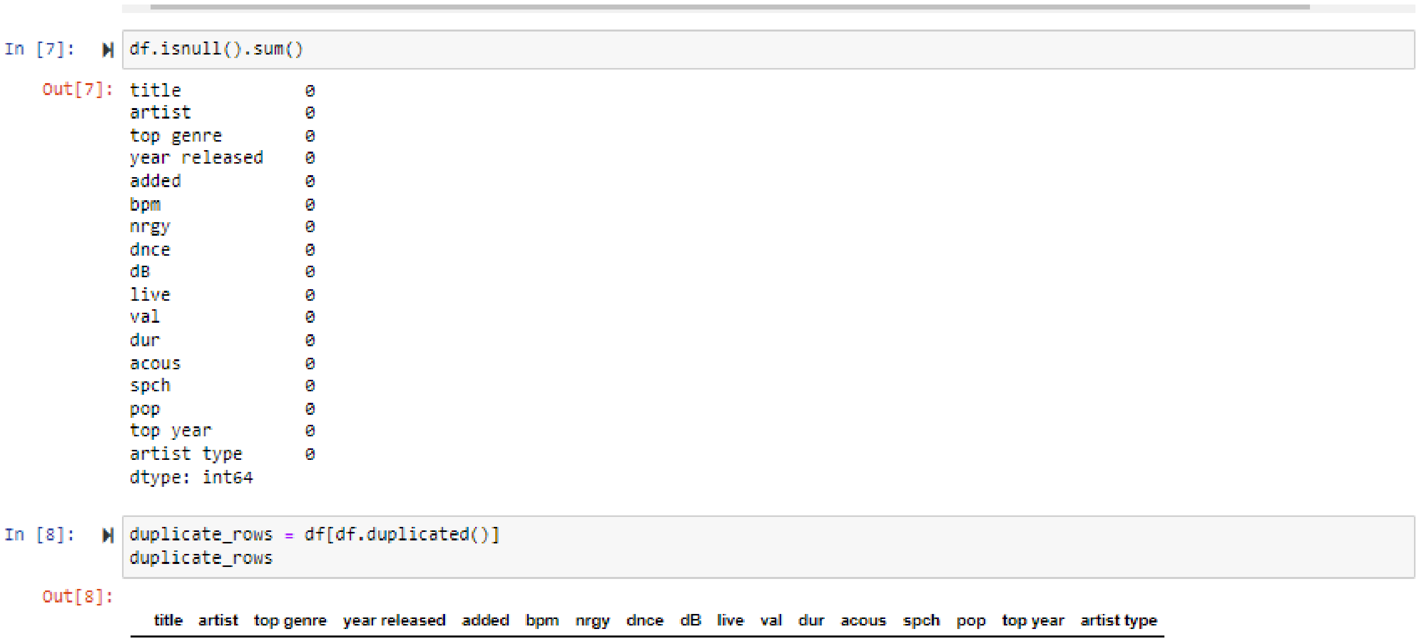


```
In [6]:  df.describe()
Out[6]:
```

	year released	bpm	nrgy	dnce	dB	live	val	dur	acous	spch	pop	
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.00000	1000.000000	1000.000000	10
mean	2014.390000	121.262000	69.502000	66.876000	-5.663000	17.911000	50.901000	220.406000	14.36900	10.064000	74.840000	20
std	3.241359	26.238022	15.961415	13.121921	2.025224	13.431511	21.563399	39.927677	19.45403	9.276743	8.807836	
min	1975.000000	65.000000	6.000000	19.000000	-18.000000	2.000000	4.000000	113.000000	0.00000	2.000000	35.000000	20
25%	2012.000000	100.000000	59.000000	59.000000	-7.000000	9.000000	35.000000	197.000000	2.00000	4.000000	70.000000	20
50%	2014.000000	122.000000	71.000000	68.000000	-5.000000	12.000000	50.500000	216.000000	6.00000	6.000000	76.000000	20
75%	2017.000000	134.000000	81.250000	75.000000	-4.000000	23.000000	68.000000	237.000000	19.00000	12.000000	81.000000	20
max	2021.000000	206.000000	98.000000	96.000000	-1.000000	83.000000	97.000000	688.000000	98.00000	53.000000	95.000000	20

I have used describe() function to view the basic statistic summary of the dataset. It contains the count, mean, standard deviation, minimum, maximum value of the columns

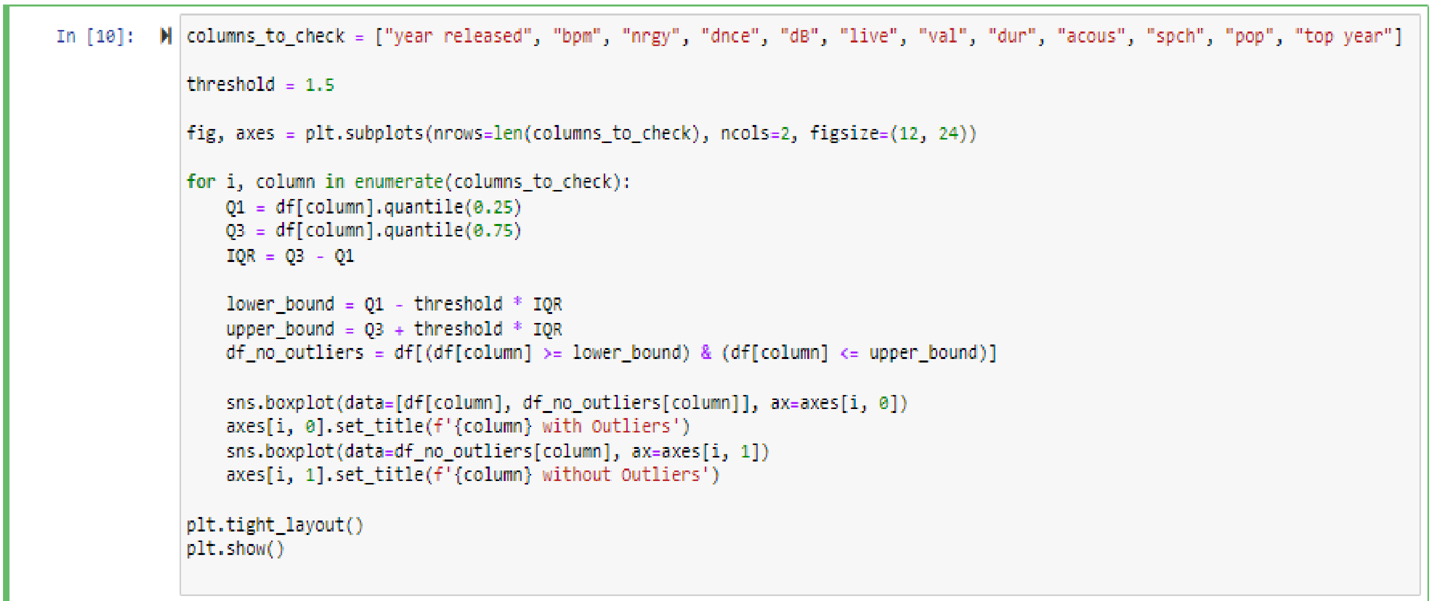
3. **Are there any categorical variables and missing values? If so, print it.**



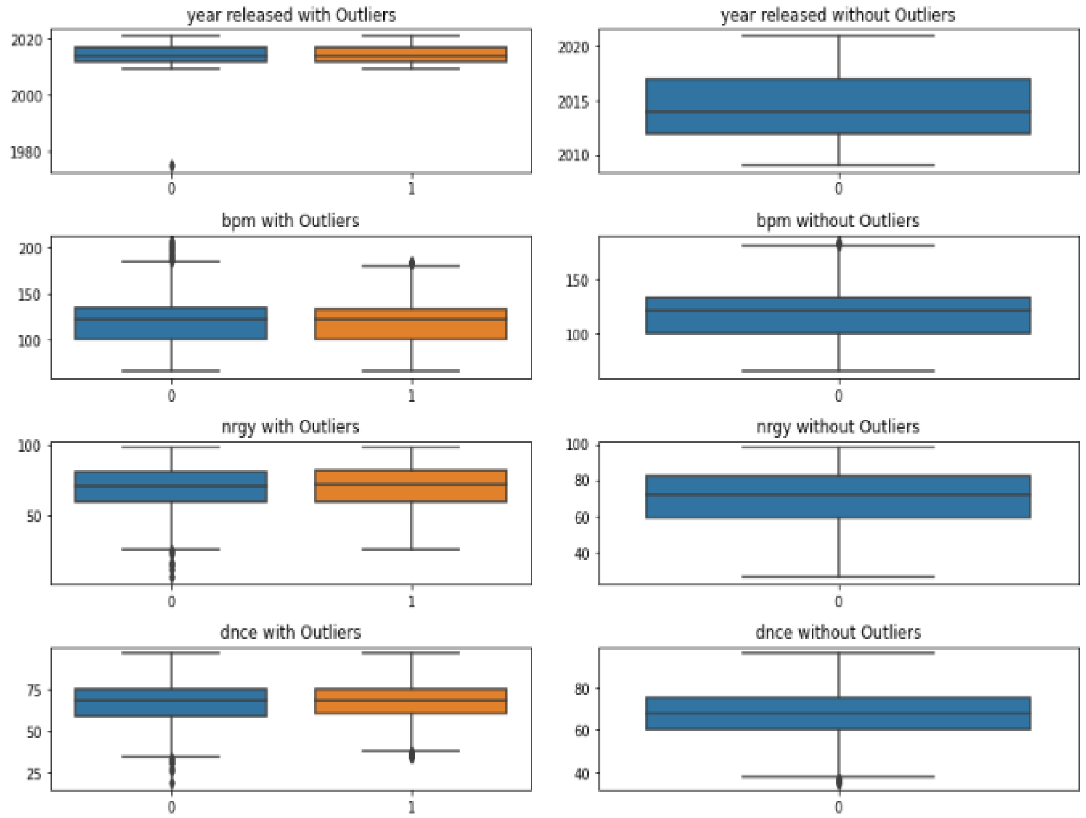
```
In [7]:  df.isnull().sum()

Out[7]:  title            0
         artist           0
         top genre        0
         year released    0
         added            0
         bpm              0
         nrgy             0
         dnce             0
         dB               0
         live             0
         val              0
         dur              0
         acous            0
         spch             0
         pop              0
         top year         0
         artist type      0
         dtype: int64

In [8]:  duplicate_rows = df[df.duplicated()]
         duplicate_rows

Out[8]:
           title  artist  top genre  year released  added  bpm  nrgy  dnce  dB  live  val  dur  acous  spch  pop  top year  artist type
```

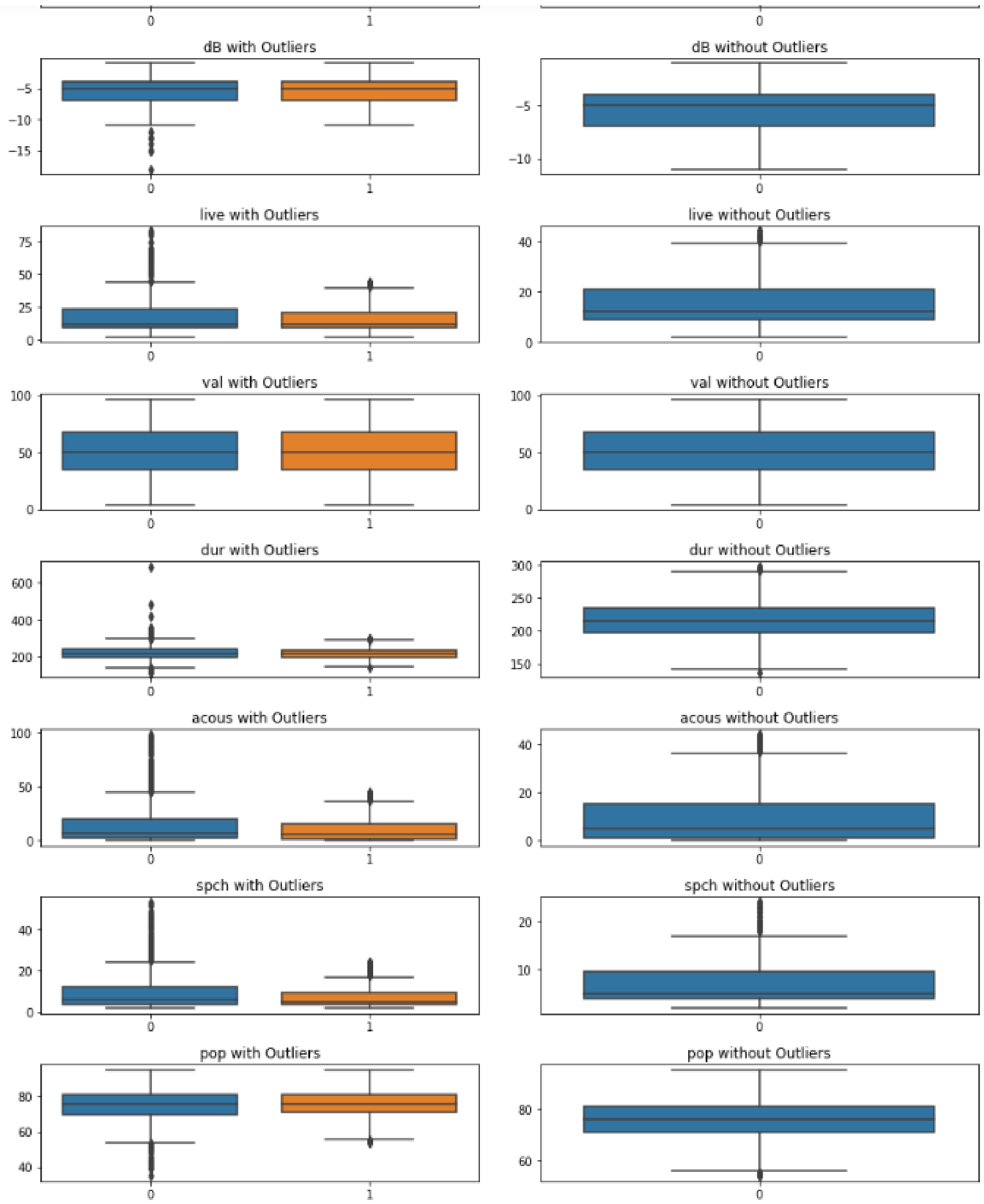
I have used df.isnull().sum() to view the missing values in the dataset. There is no missing values or duplicate values in the dataset.

4. Are there any outliers in the data? If so, use box plots, histograms and visualize.



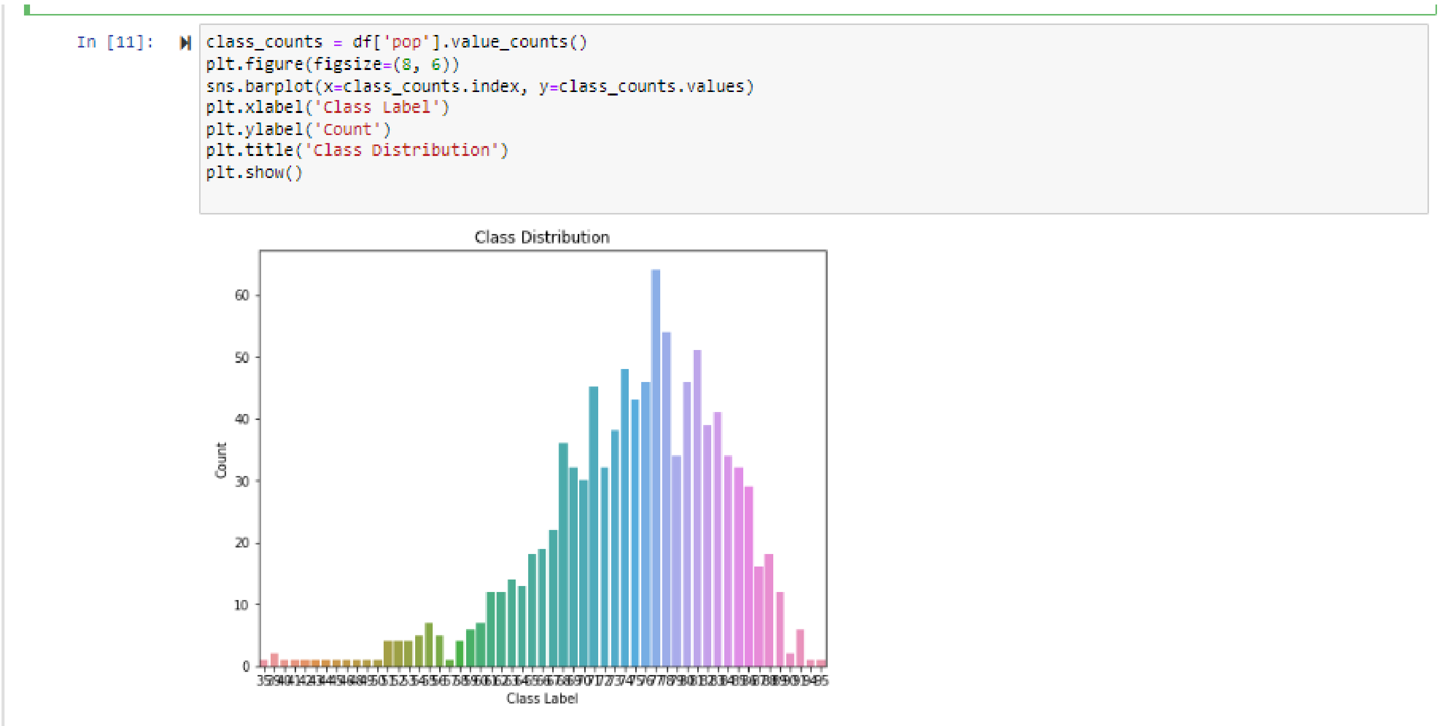
```
In [10]:  columns_to_check = ["year released", "bpm", "nrgy", "dnce", "dB", "live", "val", "dur", "acous", "spch", "pop", "top year"]

          threshold = 1.5

          fig, axes = plt.subplots(nrows=len(columns_to_check), ncols=2, figsize=(12, 24))

          for i, column in enumerate(columns_to_check):
              Q1 = df[column].quantile(0.25)
              Q3 = df[column].quantile(0.75)
              IQR = Q3 - Q1

              lower_bound = Q1 - threshold * IQR
              upper_bound = Q3 + threshold * IQR
              df_no_outliers = df[(df[column] >= lower_bound) & (df[column] <= upper_bound)]

              sns.boxplot(data=[df[column], df_no_outliers[column]], ax=axes[i, 0])
              axes[i, 0].set_title(f'{column} with Outliers')
              sns.boxplot(data=df_no_outliers[column], ax=axes[i, 1])
              axes[i, 1].set_title(f'{column} without Outliers')

          plt.tight_layout()
          plt.show()
```





The outliers present in the dataset are found and removed using the IQR. The outliers can be found using the boxplots.

5. **Is the data balanced or imbalanced? Visualize.**



```
In [11]:  class_counts = df['pop'].value_counts()
          plt.figure(figsize=(8, 6))
          sns.barplot(x=class_counts.index, y=class_counts.values)
          plt.xlabel('Class Label')
          plt.ylabel('Count')
          plt.title('Class Distribution')
          plt.show()
```

The given dataset is imbalanced. This can be found using the class distribution of the dataset.

6. **What is the target variable (if any).**

The target variable is ‘pop’ which means popularity.

7. **What are the units of measurement for numerical columns? (Example: time, currency, date, distance)**

Units of measurements:

- top genre
- year released
- added
- bpm
- nrgy
- dnce
- dB

- live
- val
- dur
- acous
- spch
- pop
- top year
- artist type

8. Do you have domain clarification? Brief it.

Domain: Spotify Music Analysis

Domain Description: Spotify music analysis involves working with data related to music tracks, albums, artists, and user interactions on the Spotify music streaming platform. This domain focuses on extracting insights, patterns, and features from music data to enhance user experiences, create personalized playlists, recommend songs, and gain a better understanding of music preferences and trends.

Key Concepts in the Domain:

- **Music Tracks:** Individual songs available on Spotify, each with its own set of attributes and features.
- **Albums:** Collections of music tracks released by artists or bands.
- **Artists:** Musicians or music groups who create music available on Spotify.
- **User Interactions:** Actions taken by Spotify users, such as listening to songs, creating playlists, following artists, and rating tracks.

- **Audio Features:** Quantitative measures of music tracks, including features like tempo (BPM), energy, danceability, loudness (dB), and acousticness, used for analysis and recommendation.
- **Popularity:** Metrics indicating the popularity of songs, albums, or artists on the platform, often based on metrics like streaming counts or user ratings.
- **Playlists:** Curated collections of music tracks created by users or Spotify's recommendation algorithms.

Applications: Spotify music analysis can be applied for various purposes, including:

- Personalized music recommendations based on user preferences.
- Creating and curating playlists.
- Identifying music trends and emerging artists.
- Understanding user engagement and behaviour on the platform.
- Enhancing music discovery and exploration.

9. Are there any time-based trends or patterns?

Time-based trends and patterns are common in music consumption and the music industry as a whole. Spotify, as a music streaming platform, captures a wealth of data related to user interactions with music over time. Analysing these data can reveal various time-based trends and patterns. Here are some examples:

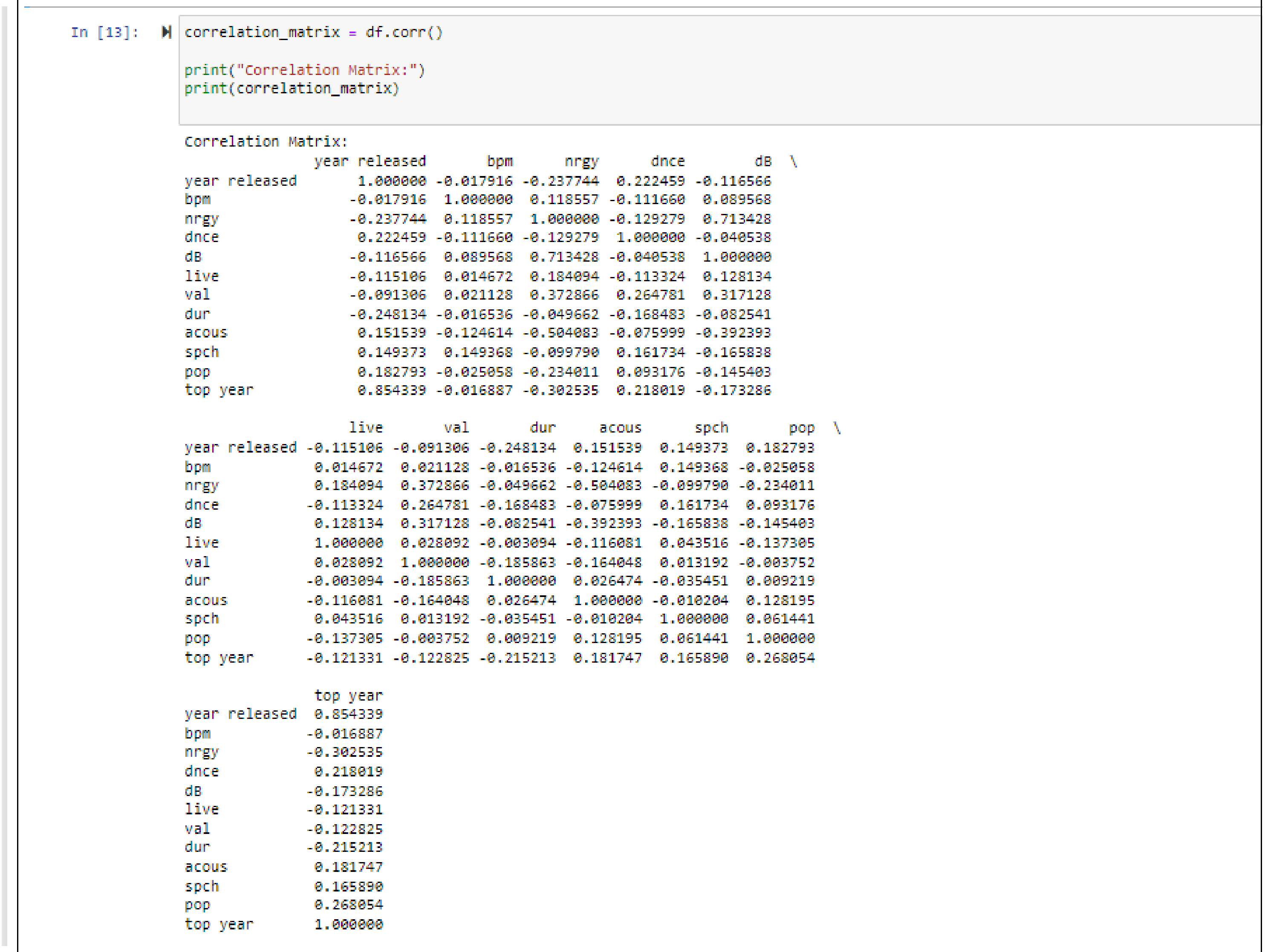
- **Seasonal Listening Trends:** Music consumption often follows seasonal patterns. For example, you might observe an increase in holiday music streaming during the winter holiday season or more upbeat and energetic music during the summer months. Seasonal trends can impact the popularity of specific genres or songs at different times of the year.

- **Release Day Trends:** New music releases, especially albums and singles from popular artists, can generate spikes in streaming activity on their release days. Fans eagerly listen to and share new releases, resulting in noticeable daily trends.
- **Day-of-the-Week Trends:** Listening behaviour can vary depending on the day of the week. For instance, weekdays might see more streaming during commuting hours, while weekends could witness different patterns as people have more leisure time.
- **Time-of-Day Trends:** Users may have distinct music preferences and listening habits depending on the time of day. For example, energetic music might be more popular in the morning, while relaxing or instrumental tracks could be favoured in the evening.
- **Playlist Trends:** Spotify's playlist features, such as Daily Mixes or Discover Weekly, often update on specific days of the week. This can lead to trends in user engagement with these playlists on particular days.
- **Event-Related Trends:** Major events, holidays, festivals, or award shows can influence music trends. For example, songs featured in a popular TV series or commercials may experience increased streaming after the show airs.
- **Artist Milestones:** When an artist releases new albums or celebrates career milestones (e.g., anniversaries), there may be a surge in listenership related to that artist's catalog.
- **Global and Regional Trends:** Trends can vary by region and country. Different cultures and geographic locations can have unique music preferences and listening habits.

- Analysing these time-based trends and patterns can help Spotify and other music platforms make informed decisions about playlist curation, content recommendations, and promotional strategies. It can also provide

- valuable insights to artists, record labels, and the music industry as a whole to better understand listener behaviour and adapt to changing preferences over time.

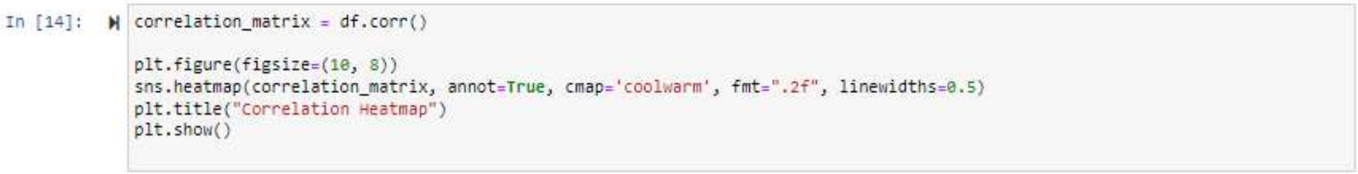
10.Are there any correlations between variables? Calculate correlations.



In [13]:

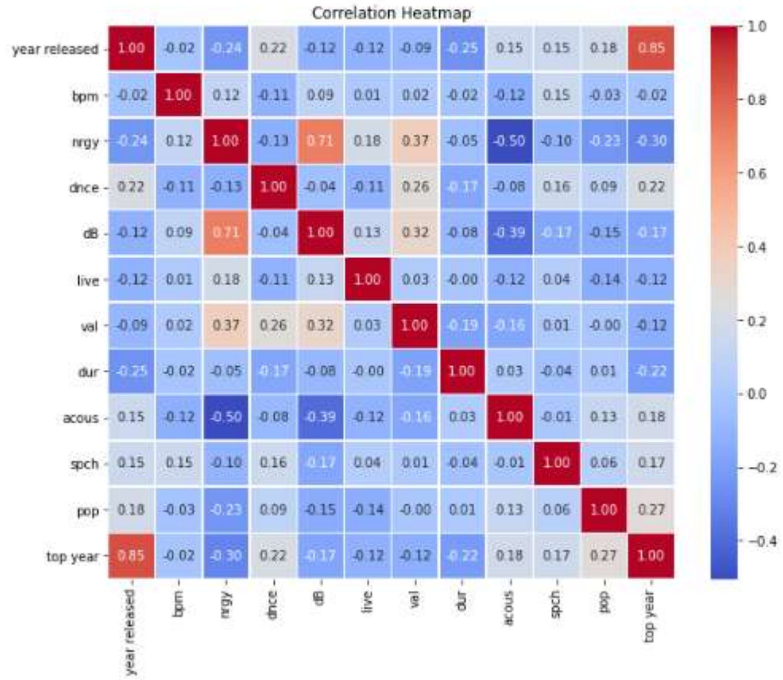
```
correlation_matrix = df.corr()

print("Correlation Matrix:")
print(correlation_matrix)
```

```
Correlation Matrix:
              year released       bpm      nrgy      dnce        dB  \
year released      1.000000 -0.017916 -0.237744  0.222459 -0.116566   
bpm               -0.017916  1.000000  0.118557 -0.111660  0.089568   
nrgy              -0.237744  0.118557  1.000000 -0.129279  0.713428   
dnce               0.222459 -0.111660 -0.129279  1.000000 -0.040538   
dB                -0.116566  0.089568  0.713428 -0.040538  1.000000   
live              -0.115106  0.014672  0.184094 -0.113324  0.128134   
val               -0.091306  0.021128  0.372866  0.264781  0.317128   
dur               -0.248134 -0.016536 -0.049662 -0.168483 -0.082541   
acous              0.151539 -0.124614 -0.504083 -0.075999 -0.392393   
spch               0.149373  0.149368 -0.099790  0.161734 -0.165838   
pop                0.182793 -0.025058 -0.234011  0.093176 -0.145403   
top year           0.854339 -0.016887 -0.302535  0.218019 -0.173286   

                   live       val       dur     acous      spch       pop  \
year released -0.115106 -0.091306 -0.248134  0.151539  0.149373  0.182793   
bpm            0.014672  0.021128 -0.016536 -0.124614  0.149368 -0.025058   
nrgy           0.184094  0.372866 -0.049662 -0.504083 -0.099790 -0.234011   
dnce          -0.113324  0.264781 -0.168483 -0.075999  0.161734  0.093176   
dB             0.128134  0.317128 -0.082541 -0.392393 -0.165838 -0.145403   
live           1.000000  0.028092 -0.003094 -0.116081  0.043516 -0.137305   
val            0.028092  1.000000 -0.185863 -0.164048  0.013192 -0.003752   
dur           -0.003094 -0.185863  1.000000  0.026474 -0.035451  0.009219   
acous         -0.116081 -0.164048  0.026474  1.000000 -0.010204  0.128195   
spch           0.043516  0.013192 -0.035451 -0.010204  1.000000  0.061441   
pop           -0.137305 -0.003752  0.009219  0.128195  0.061441  1.000000   
top year      -0.121331 -0.122825 -0.215213  0.181747  0.165890  0.268054   

               top year  
year released  0.854339  
bpm           -0.016887  
nrgy          -0.302535  
dnce           0.218019  
dB            -0.173286  
live          -0.121331  
val           -0.122825  
dur           -0.215213  
acous          0.181747  
spch           0.165890  
pop            0.268054  
top year       1.000000  
```



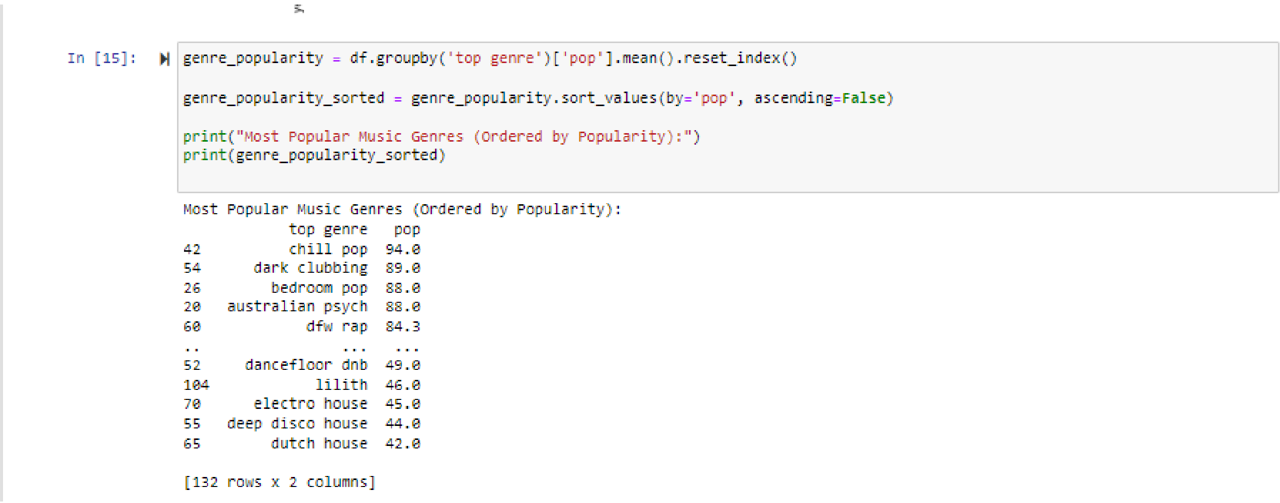
In [14]:

```
correlation_matrix = df.corr()

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title("Correlation Heatmap")
plt.show()
```



The correlation between the variables can be found by correlation matrix and heatmap. The columns top year and year released are strongly correlated with score of 0.85 followed by dB(loudness) and nrgy(energy) with the score of 0.71.

11. Which music genres are the most popular among users?



In [15]:

```
genre_popularity = df.groupby('top genre')['pop'].mean().reset_index()

genre_popularity_sorted = genre_popularity.sort_values(by='pop', ascending=False)

print("Most Popular Music Genres (Ordered by Popularity):")
print(genre_popularity_sorted)
```

```
Most Popular Music Genres (Ordered by Popularity):
             top genre   pop
42           chill pop  94.0
54       dark clubbing  89.0
26         bedroom pop  88.0
20     australian psych  88.0
60             dfw rap  84.3
..                 ...   ...
52       dancefloor dnb  49.0
104            lilith  46.0
70       electro house  45.0
55    deep disco house  44.0
65         dutch house  42.0

[132 rows x 2 columns]
```

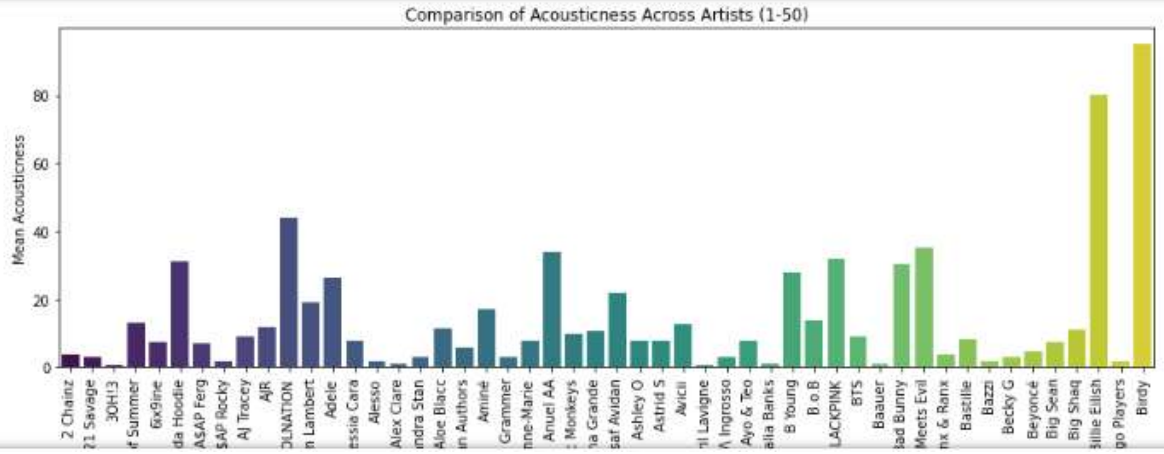
The most popular Genre is chill pop with pop score of 94.

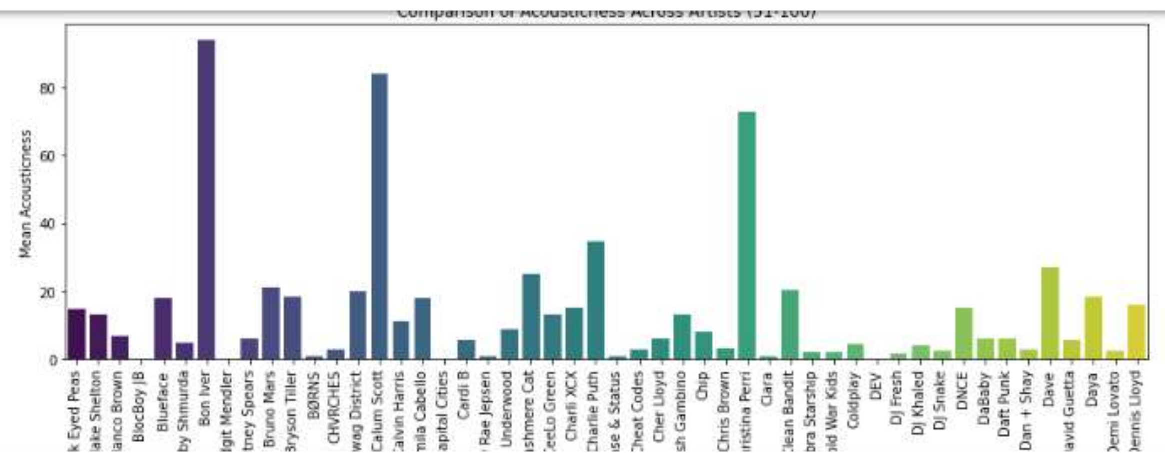
12. What features (e.g., acousticness, danceability) are associated with popular songs?

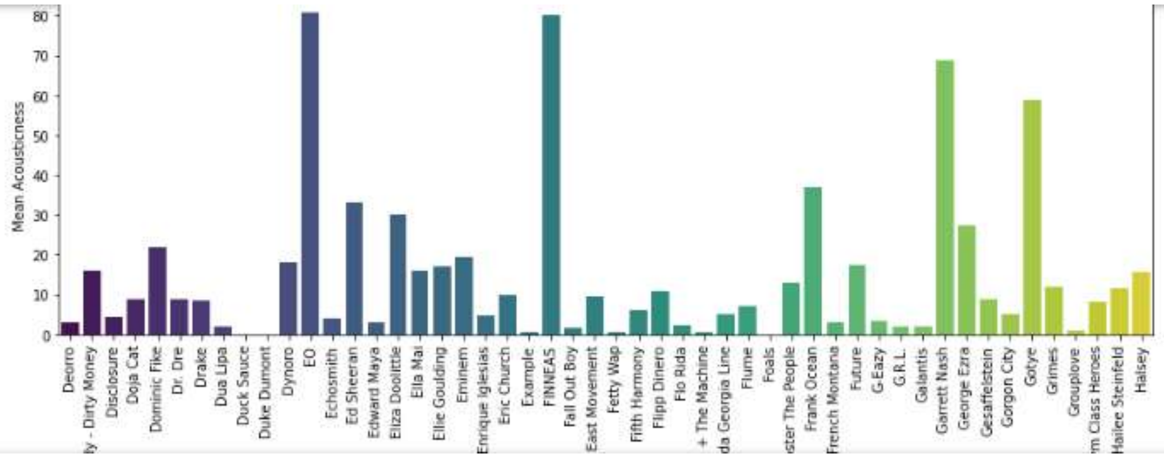
According the Heatmap which is used to find correlation the important factors associated with the popular songs are

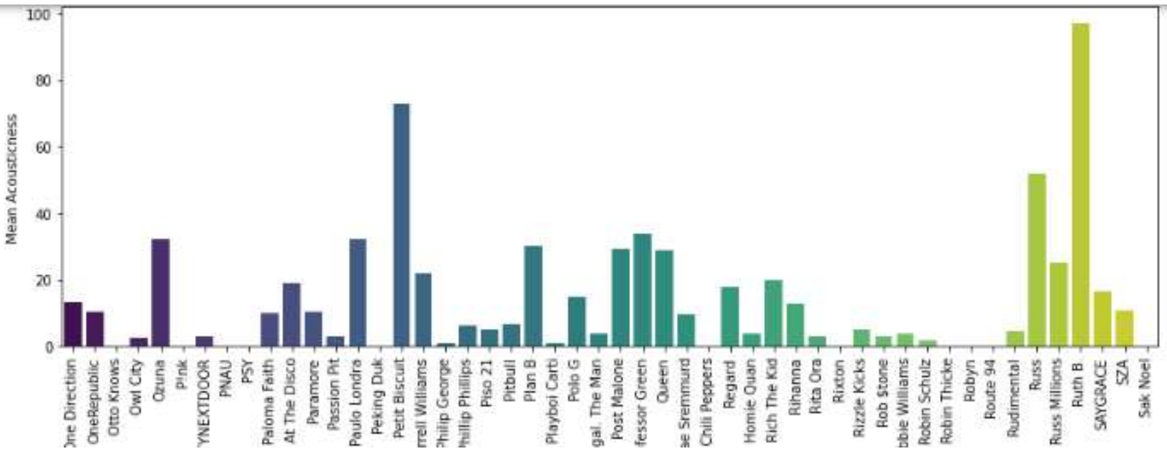
- Year Released
- Acousticness

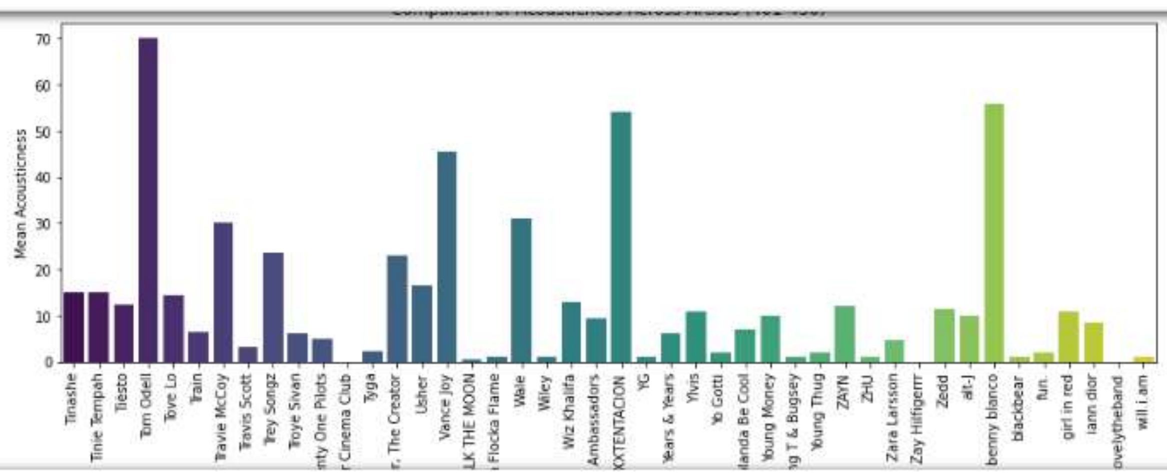
13.**How do different artists' catalogues compare in terms of acoustic features?**

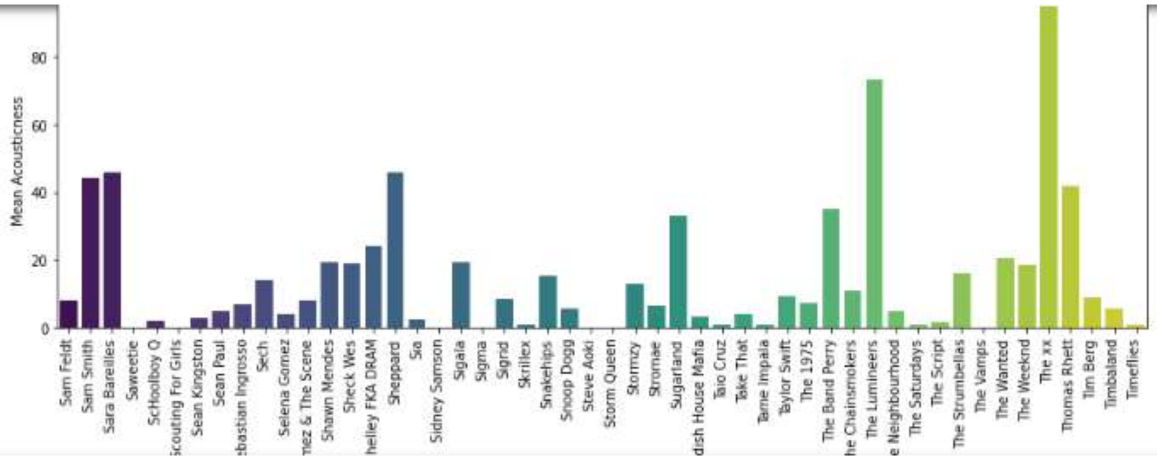


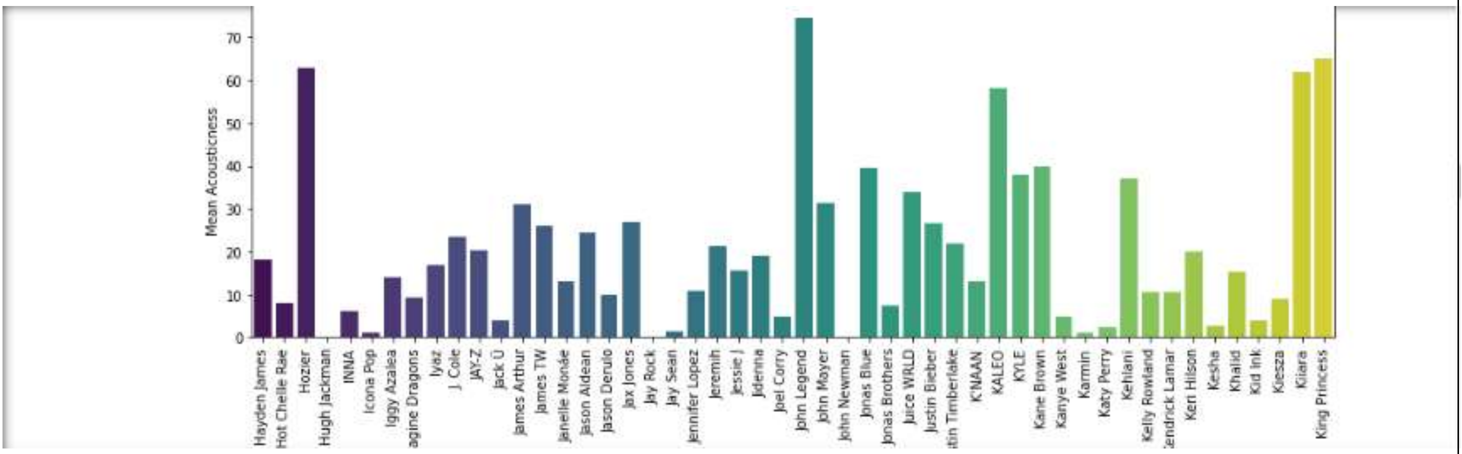


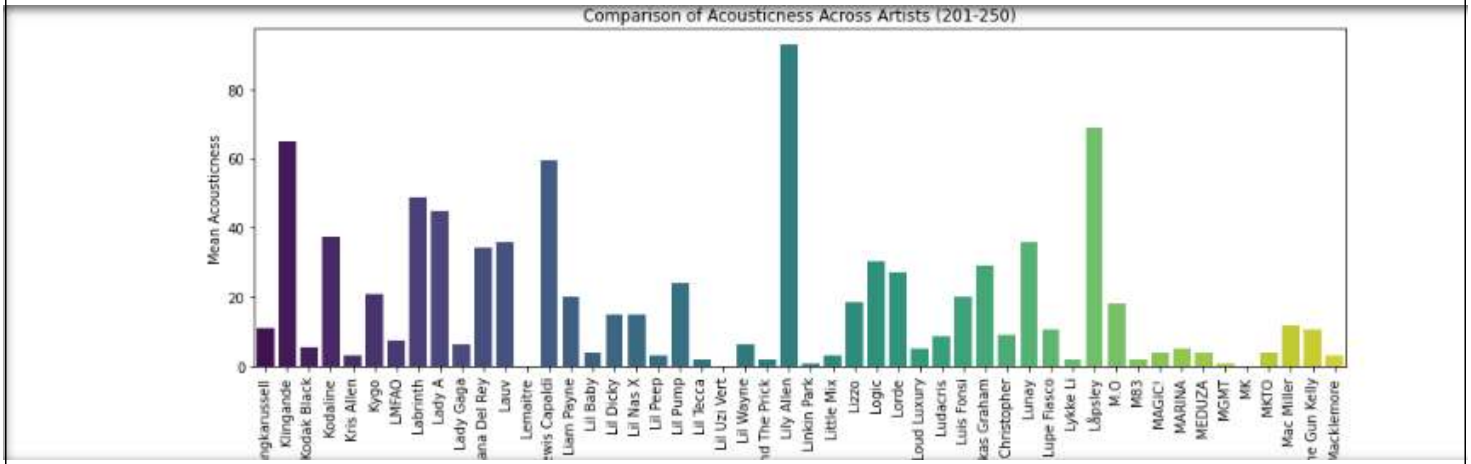


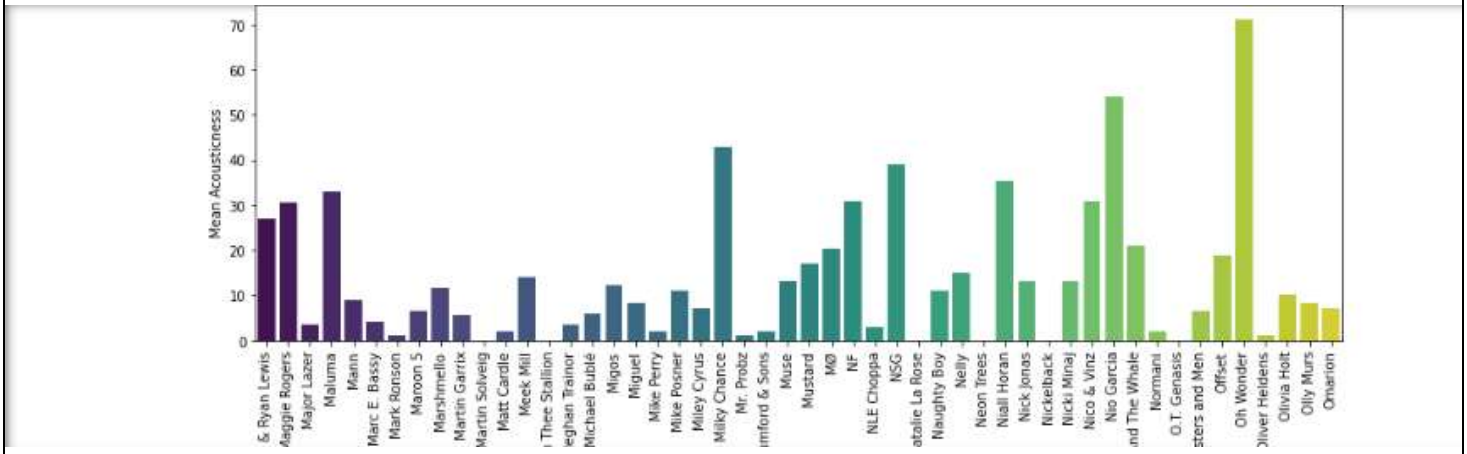

Mean Acousticness
0
20
40
60
80
100
One Direction
OneRepublic
Otto Knows
Owl City
Ozuna
Pink
YNEXTDOOR
PNAU
PSY
Paloma Faith
At The Disco
Paramore
Passion Pit
Paulo Londra
Peking Duk
Petit Biscuit
rrell Williams
Philip George
Phillip Phillips
Piso 21
Pitbull
Plan B
Playboi Carti
Polo G
gal. The Man
Post Malone
fessor Green
Queen
ae Sremmurd
Chili Peppers
Regard
Homie Quan
Rich The Kid
Rihanna
Rita Ora
Rixton
Rizzle Kicks
Rob $tone
bbie Williams
Robin Schulz
Robin Thicke
Robyn
Route 94
Rudimental
Russ
Russ Millions
Ruth B
SAYGRACE
SZA
Sak Noel


Comparison of Acousticness Across Artists (401-450)
Mean Acousticness
0
10
20
30
40
50
60
70
Tinashe
Tinie Tempah
Tiesto
Tom Odell
Tove Lo
Train
Travie McCoy
Travis Scott
Trey Songz
Troye Sivan
nty One Pilots
r Cinema Club
Tyga
r, The Creator
Usher
Vance Joy
LK THE MOON
Flocka Flame
Wale
Wiley
Wiz Khalifa
Ambassadors
XXTENTACION
YG
Years & Years
Ylvis
Yo Gotti
landa Be Cool
Young Money
ng T & Bugsey
Young Thug
ZAYN
ZHU
Zara Larsson
Zay Hilfigerrr
Zedd
alt-J
benny blanco
blackbear
fun.
girl in red
iann dior
ovelytheband
will.i.am


Mean Acousticness
0
20
40
60
80
Sam Feldt
Sam Smith
Sara Bareilles
Saweetie
ScHoolboy Q
Scouting For Girls
Sean Kingston
Sean Paul
ebastian Ingrosso
Sech
Selena Gomez
mez & The Scene
Shawn Mendes
Sheck Wes
helley FKA DRAM
Sheppard
Sia
Sidney Samson
Sigala
Sigma
Sigrid
Skrillex
Snakehips
Snoop Dogg
Steve Aoki
Storm Queen
Stormzy
Stromae
Sugarland
dish House Mafia
Taio Cruz
Take That
Tame Impala
Taylor Swift
The 1975
The Band Perry
he Chainsmokers
The Lumineers
e Neighbourhood
The Saturdays
The Script
The Strumbellas
The Vamps
The Wanted
The Weeknd
The xx
Thomas Rhett
Tim Berg
Timbaland
Timeflies

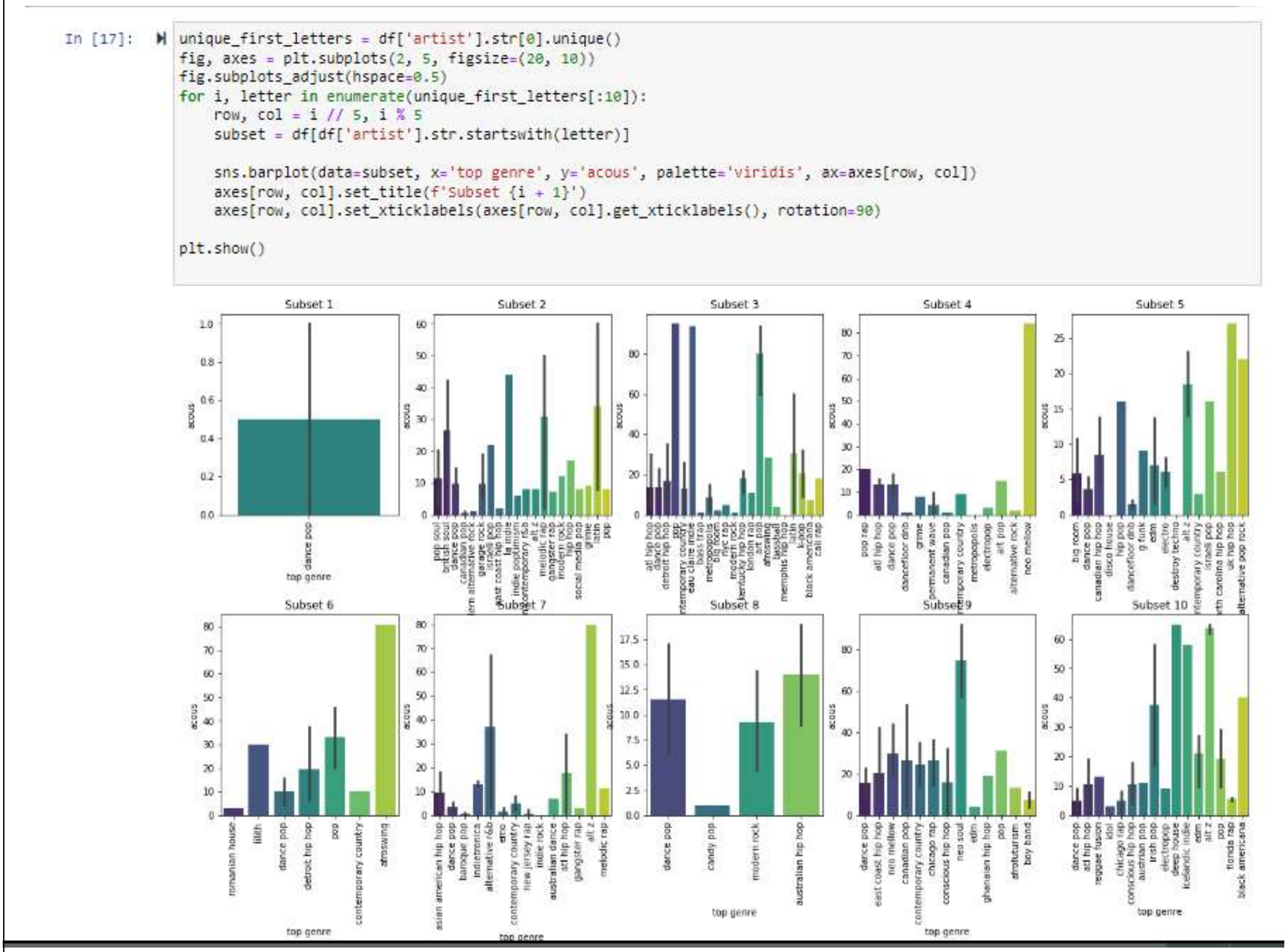



Comparison of Acousticness Across Artists (201-250)



The above graphs show the acousticness across various artists

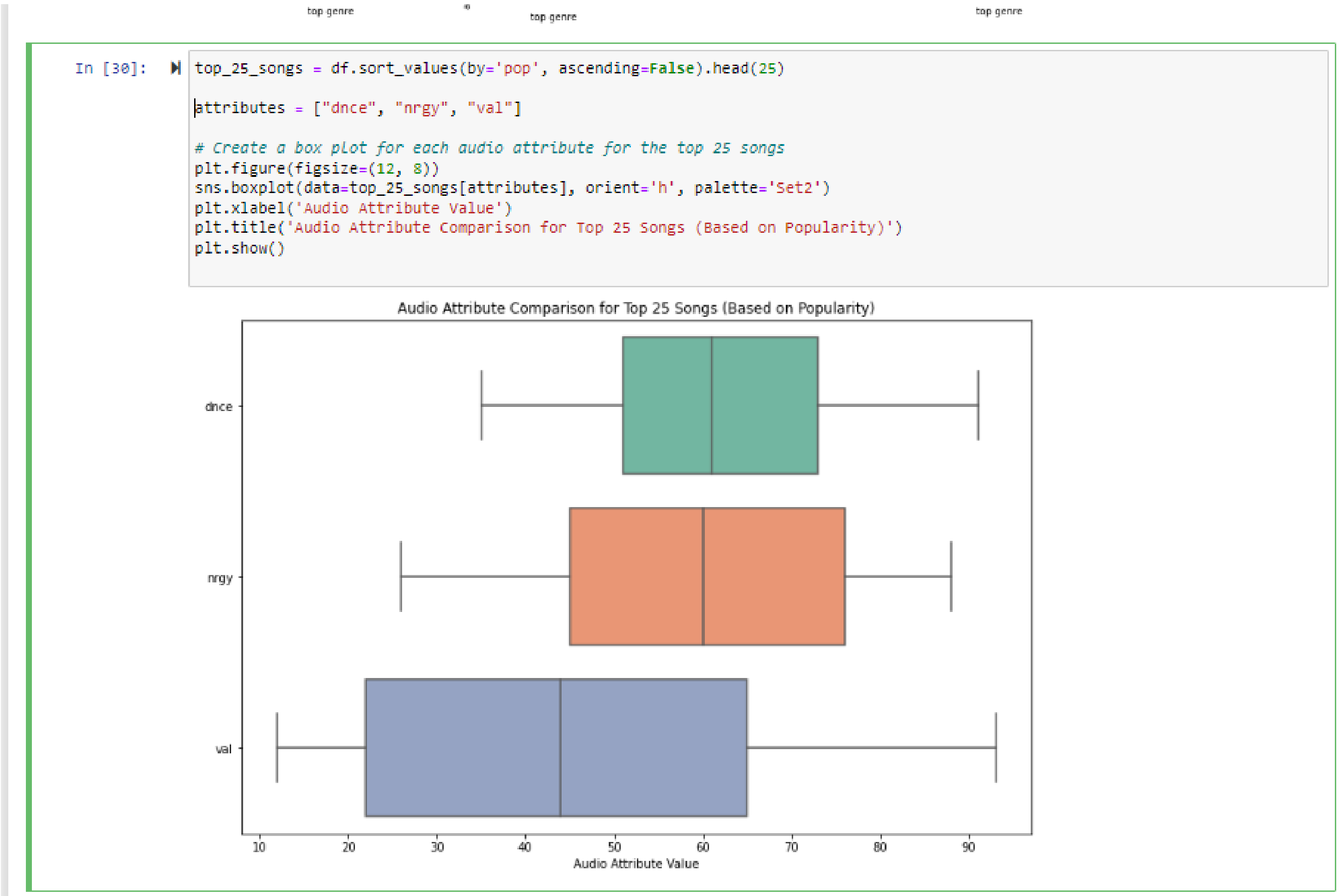
14. **Are there trends in audio feature preferences across different genres**?



```
In [17]:  unique_first_letters = df['artist'].str[0].unique()
          fig, axes = plt.subplots(2, 5, figsize=(20, 10))
          fig.subplots_adjust(hspace=0.5)
          for i, letter in enumerate(unique_first_letters[:10]):
              row, col = i // 5, i % 5
              subset = df[df['artist'].str.startswith(letter)]

              sns.barplot(data=subset, x='top genre', y='acous', palette='viridis', ax=axes[row, col])
              axes[row, col].set_title(f'Subset {i + 1}')
              axes[row, col].set_xticklabels(axes[row, col].get_xticklabels(), rotation=90)

          plt.show()
```

These are the trends in audio preference among users across different genre.

15.Explore which audio attributes, such as danceability, energy, and valence, are commonly associated with top-charting songs.

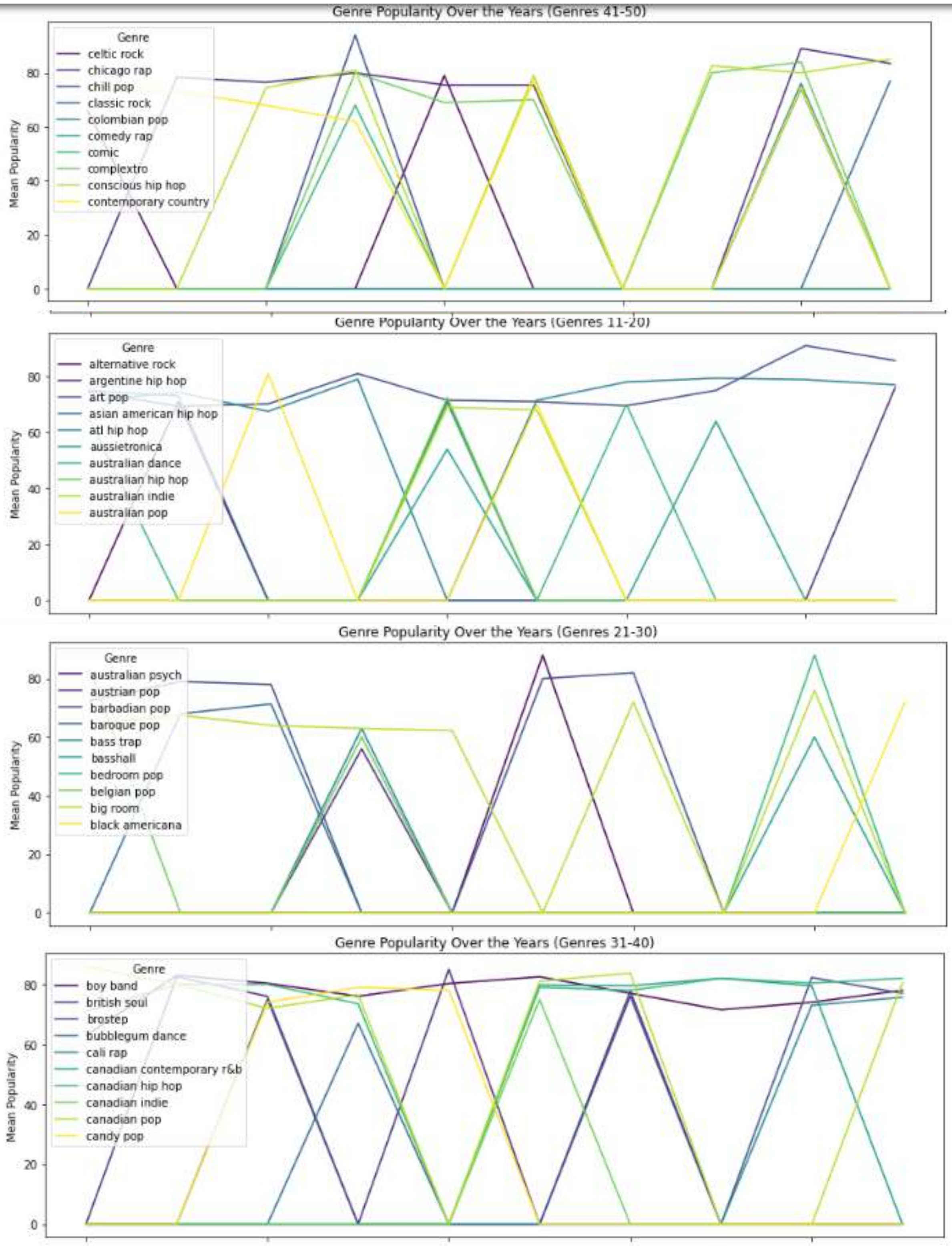


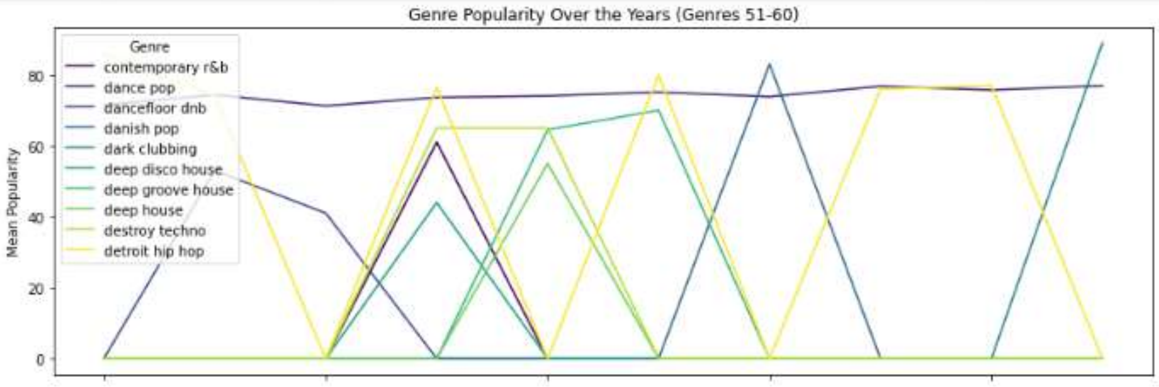
```
top_25_songs = df.sort_values(by='pop', ascending=False).head(25)

attributes = ["dnce", "nrgy", "val"]

# Create a box plot for each audio attribute for the top 25 songs
plt.figure(figsize=(12, 8))
sns.boxplot(data=top_25_songs[attributes], orient='h', palette='Set2')
plt.xlabel('Audio Attribute Value')
plt.title('Audio Attribute Comparison for Top 25 Songs (Based on Popularity)')
plt.show()
```

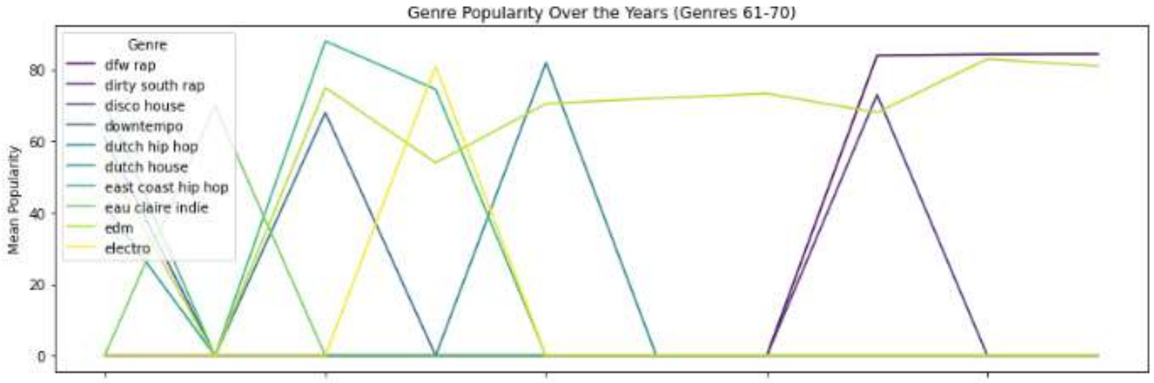
From the above graph we can know that the attributes have the following value range.

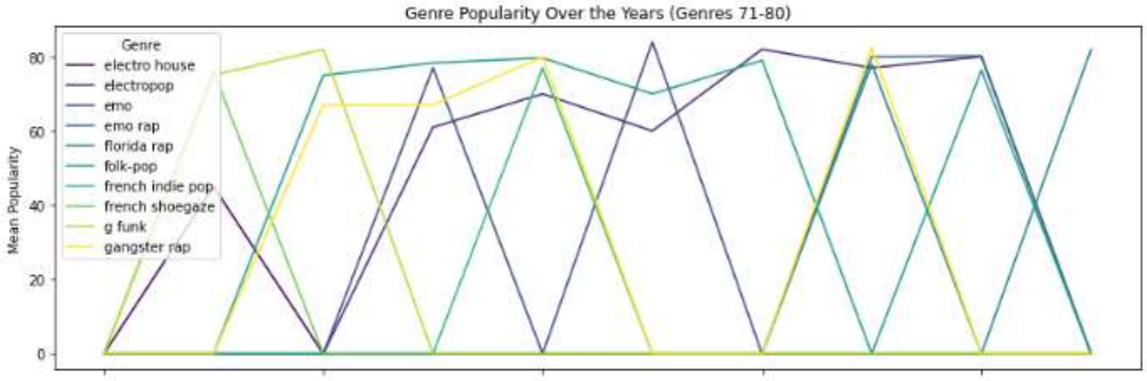
- Danceability – 50 to 72
- Energy – 45 to 75
- Valence – 22 to 65

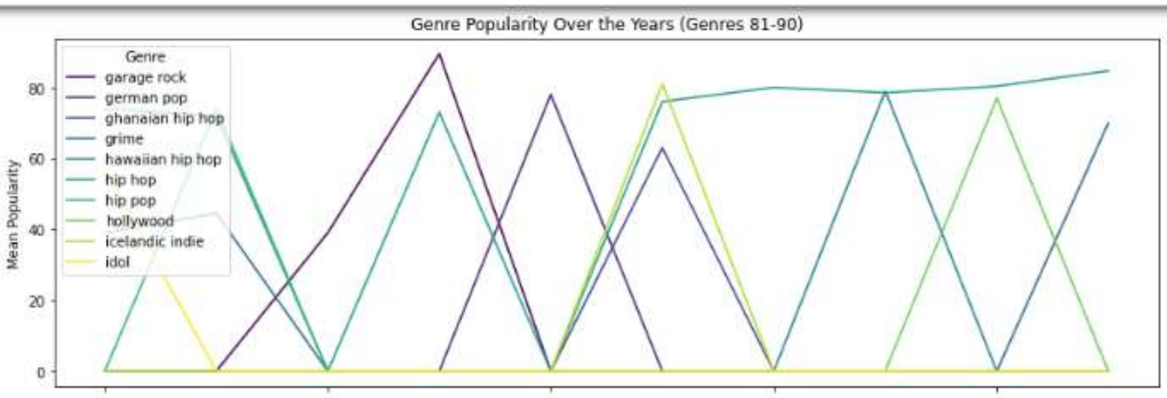
16. How have music genre preferences evolved over time?

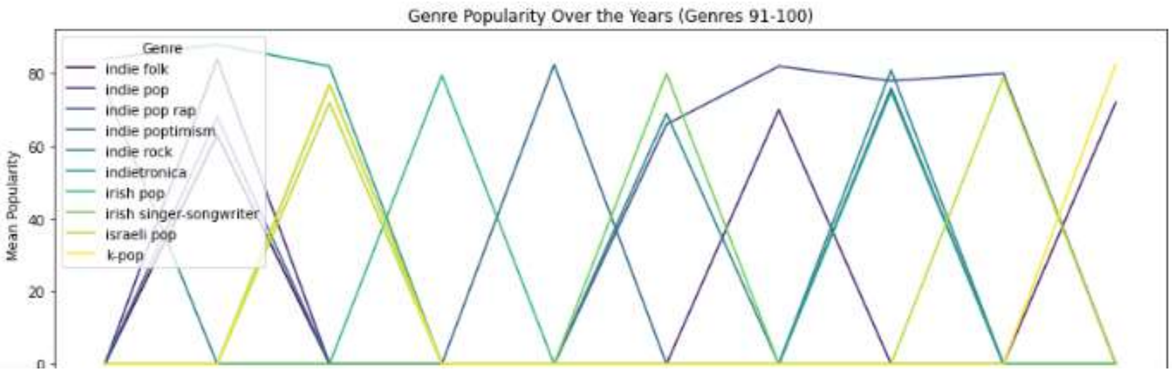


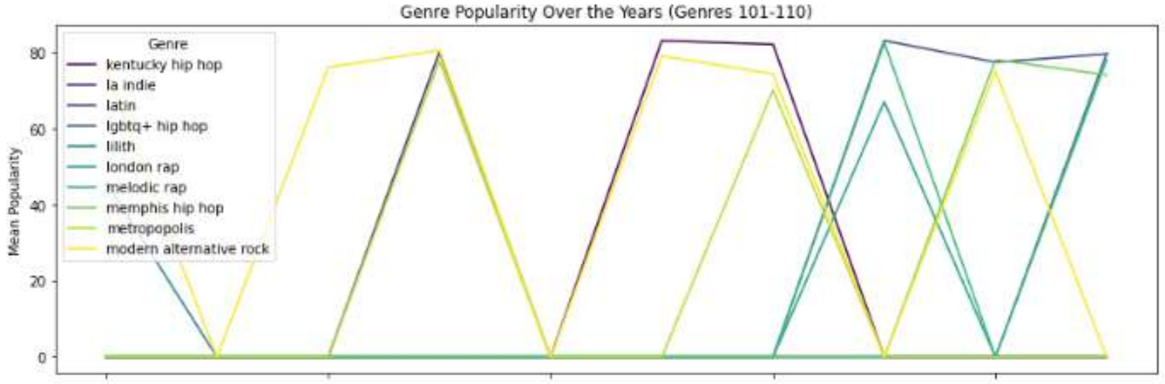

Genre Popularity Over the Years (Genres 51-60)
Mean Popularity
0
20
40
60
80
Genre
contemporary r&b
dance pop
dancefloor dnb
danish pop
dark clubbing
deep disco house
deep groove house
deep house
destroy techno
detroit hip hop


Genre Popularity Over the Years (Genres 61-70)
Mean Popularity
0
20
40
60
80
Genre
dfw rap
dirty south rap
disco house
downtempo
dutch hip hop
dutch house
east coast hip hop
eau claire indie
edm
electro


Genre Popularity Over the Years (Genres 71-80)
Mean Popularity
0
20
40
60
80
Genre
electro house
electropop
emo
emo rap
florida rap
folk-pop
french indie pop
french shoegaze
g funk
gangster rap

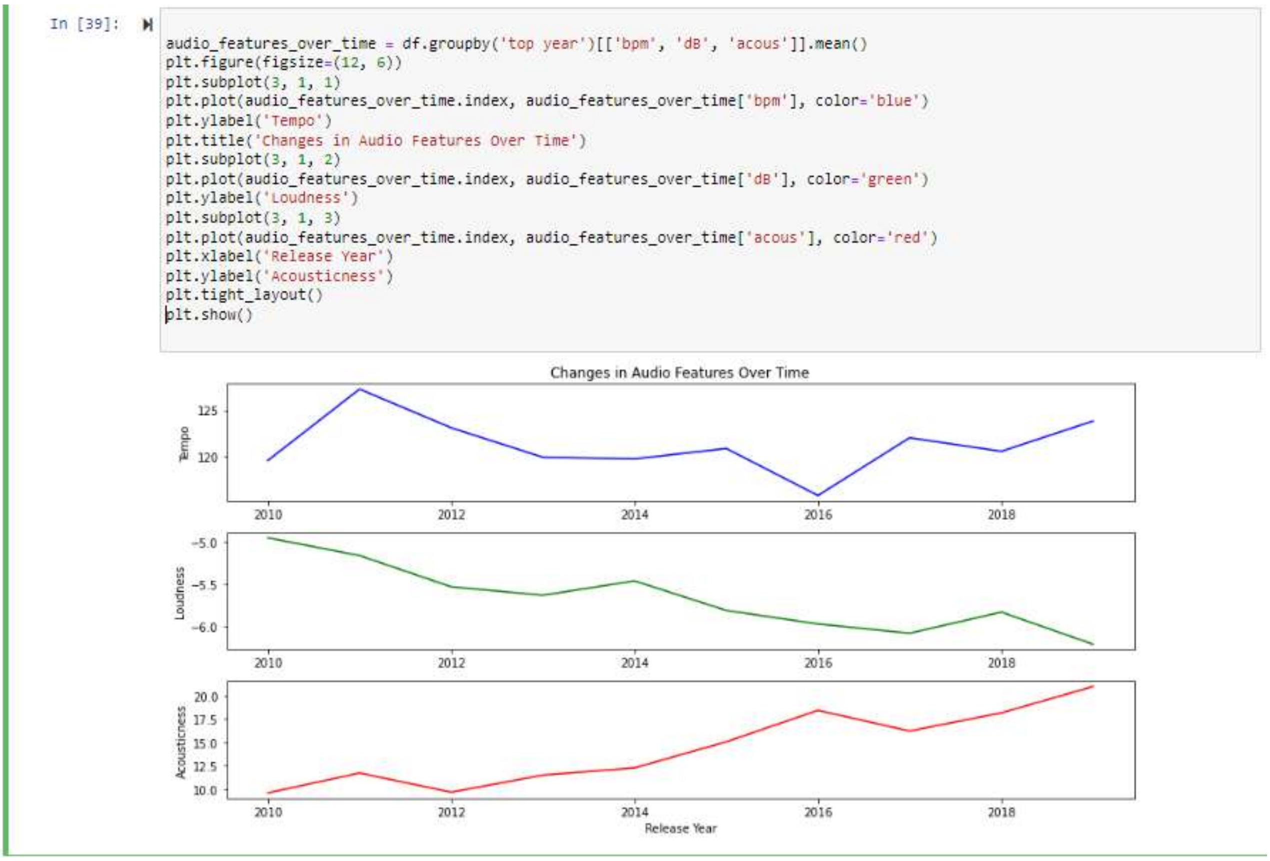

Genre Popularity Over the Years (Genres 81-90)
Mean Popularity
0
20
40
60
80
Genre
garage rock
german pop
ghanaian hip hop
grime
hawaiian hip hop
hip hop
hip pop
hollywood
icelandic indie
idol





These are the trends in various Genre over the years.

18.Are there notable changes in audio features like tempo, loudness, or Instrumentalness in songs over time?



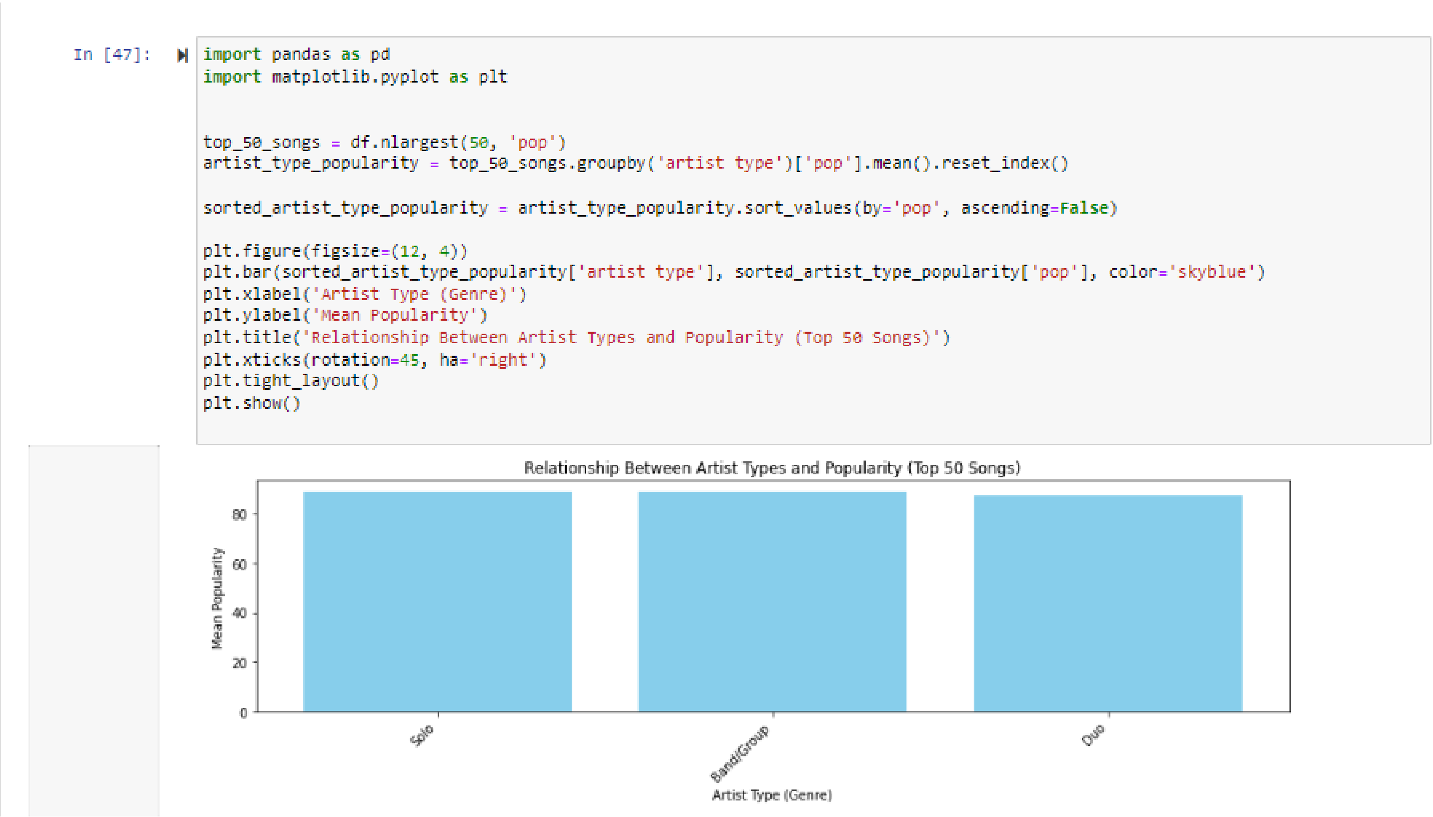
In [39]:

```
audio_features_over_time = df.groupby('top year')[['bpm', 'dB', 'acous']].mean()
plt.figure(figsize=(12, 6))
plt.subplot(3, 1, 1)
plt.plot(audio_features_over_time.index, audio_features_over_time['bpm'], color='blue')
plt.ylabel('Tempo')
plt.title('Changes in Audio Features Over Time')
plt.subplot(3, 1, 2)
plt.plot(audio_features_over_time.index, audio_features_over_time['dB'], color='green')
plt.ylabel('Loudness')
plt.subplot(3, 1, 3)
plt.plot(audio_features_over_time.index, audio_features_over_time['acous'], color='red')
plt.xlabel('Release Year')
plt.ylabel('Acousticness')
plt.tight_layout()
plt.show()
```

The change in audio features such as tempo, loudness and acousticness is shown in the above graph.

We can see the fall in Loudness and increase in Acousticness over the years.

19. **Is there relationship between the top and artist type?**



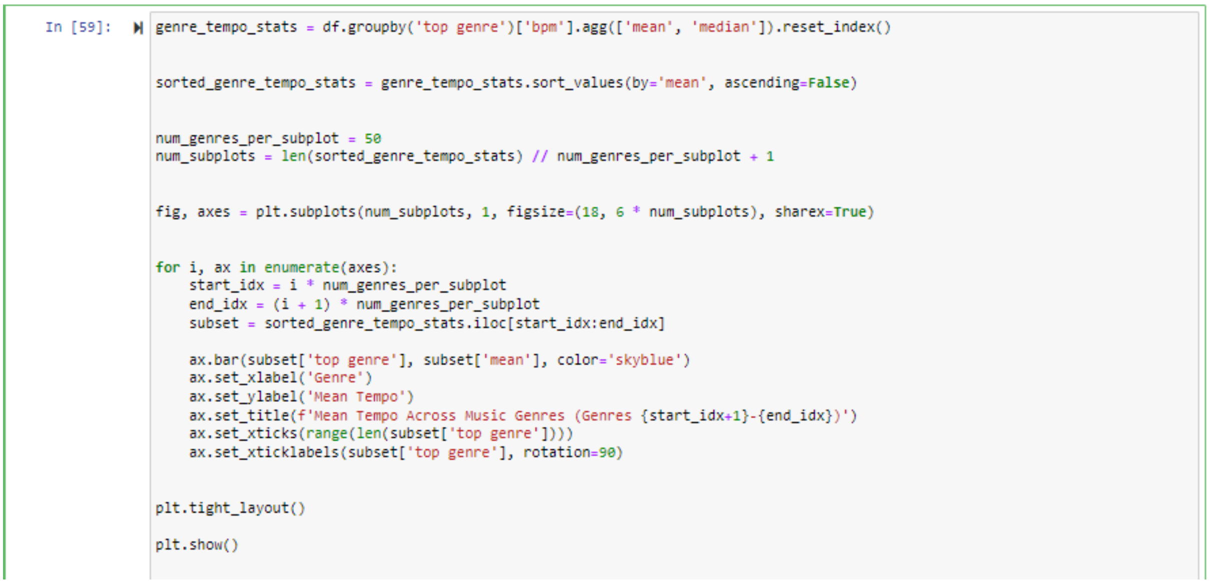
In [47]:

```
import pandas as pd
import matplotlib.pyplot as plt


top_50_songs = df.nlargest(50, 'pop')
artist_type_popularity = top_50_songs.groupby('artist type')['pop'].mean().reset_index()

sorted_artist_type_popularity = artist_type_popularity.sort_values(by='pop', ascending=False)

plt.figure(figsize=(12, 4))
plt.bar(sorted_artist_type_popularity['artist type'], sorted_artist_type_popularity['pop'], color='skyblue')
plt.xlabel('Artist Type (Genre)')
plt.ylabel('Mean Popularity')
plt.title('Relationship Between Artist Types and Popularity (Top 50 Songs)')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

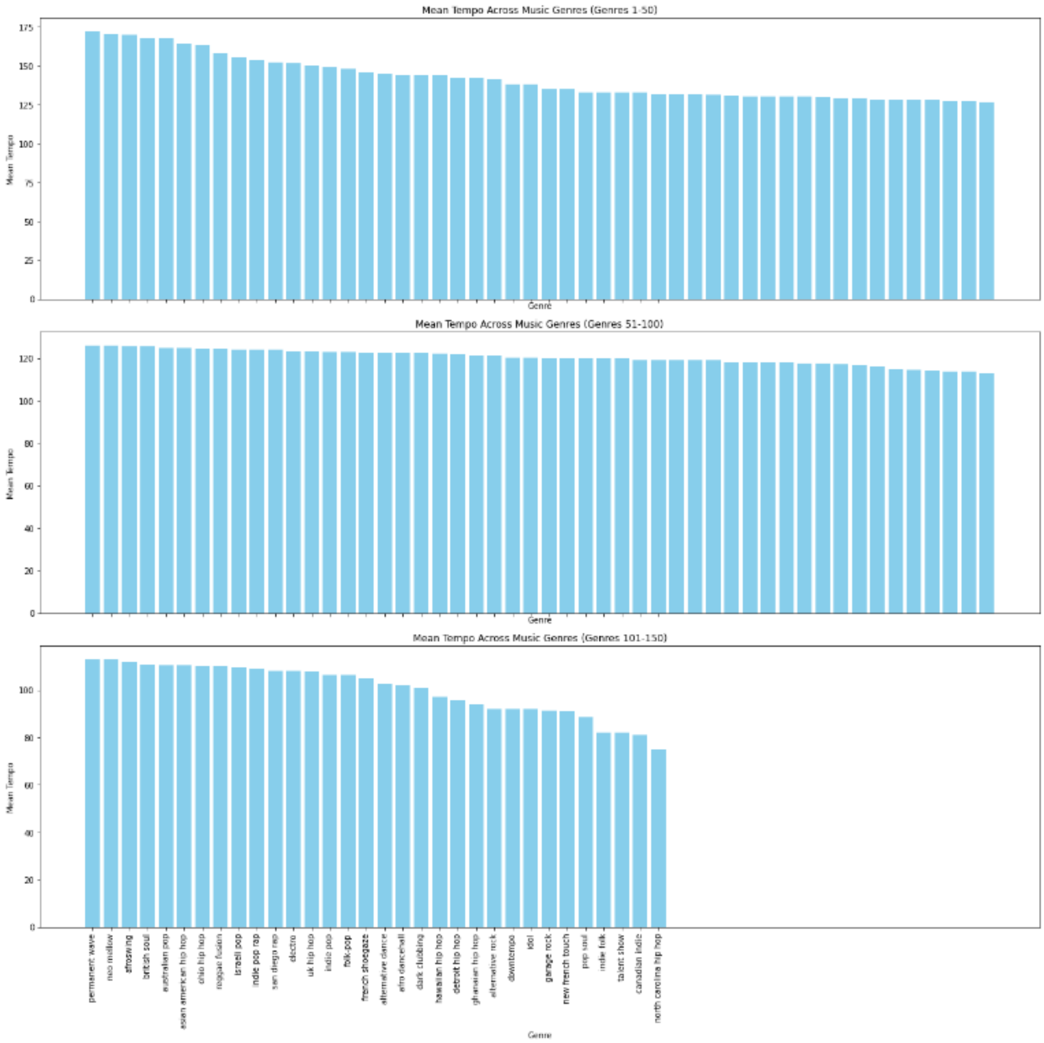
Relationship Between Artist Types and Popularity (Top 50 Songs)
Mean Popularity
0
20
40
60
80
Solo
Band/Group
Duo
Artist Type (Genre)

As we see there is no relationship between the artist type and the popularity of the music.

20.Are there tempo preferences associated with specific music genres, and can we determine the typical tempos for different genres?



```
In [59]:  genre_tempo_stats = df.groupby('top genre')['bpm'].agg(['mean', 'median']).reset_index()


           sorted_genre_tempo_stats = genre_tempo_stats.sort_values(by='mean', ascending=False)


           num_genres_per_subplot = 50
           num_subplots = len(sorted_genre_tempo_stats) // num_genres_per_subplot + 1


           fig, axes = plt.subplots(num_subplots, 1, figsize=(18, 6 * num_subplots), sharex=True)


           for i, ax in enumerate(axes):
               start_idx = i * num_genres_per_subplot
               end_idx = (i + 1) * num_genres_per_subplot
               subset = sorted_genre_tempo_stats.iloc[start_idx:end_idx]

               ax.bar(subset['top genre'], subset['mean'], color='skyblue')
               ax.set_xlabel('Genre')
               ax.set_ylabel('Mean Tempo')
               ax.set_title(f'Mean Tempo Across Music Genres (Genres {start_idx+1}-{end_idx})')
               ax.set_xticks(range(len(subset['top genre'])))
               ax.set_xticklabels(subset['top genre'], rotation=90)


           plt.tight_layout()

           plt.show()
```



The tempo preference associated with specific music genres and typical tempos for different genre has been visualized through the above graph.

Challenges and Limitation of this data set:

Data Privacy Concerns:

Spotify, like many other platforms, deals with sensitive user data. Ensuring privacy and complying with data protection regulations (such as GDPR) can be challenging.

Data Volume and Variety:

Spotify generates a massive amount of data daily due to its large user base and extensive music catalogue. Handling and analysing such diverse and voluminous data can be resource-intensive.

Data Quality:

Ensuring the accuracy and completeness of data can be a challenge. Incomplete or inaccurate data may affect the reliability of any analysis or model built on the dataset.

User Behaviour Complexity:

User behaviour on music streaming platforms is multifaceted. It includes listening habits, device preferences, interaction patterns, and more. Understanding and modelling this complex behaviour can be challenging.

Dynamic Nature of the Platform:

Spotify regularly updates its platform, introduces new features, and modifies existing ones. Keeping datasets up-to-date and adapting analyses or models to these changes can be a continuous challenge.

Limited Context:

Datasets may not always capture the full context of user interactions. For example, understanding why a user skipped a song or created a particular playlist might require additional contextual information that may not be present in the dataset.

Cold Start Problem:

Recommender systems face the cold start problem, especially with new users or newly released songs. It's challenging to make accurate recommendations when there is limited historical data for a user or a track.

Bias in Data:

Bias can exist in the data, reflecting the preferences of the majority user group. This can lead to recommendations that may not be suitable for minority users or may reinforce existing biases.

Lack of Interoperability:

Spotify data might not always be easily interoperable with other datasets or systems, limiting the potential for comprehensive analyses that involve multiple data sources.

Commercial Considerations:

Certain datasets may be limited or not available for public use due to commercial considerations. This can restrict researchers or developers from accessing certain types of data.

CONCLUSION

In navigating the vast sea of Spotify's data, our analysis has unfurled a melodic narrative, offering profound insights into user interactions and musical predilections. The rhythm of user engagement, a heartbeat coursing through the platform, reveals itself in the ebb and flow of playtime data. Discerning patterns within usage hours provides not just a temporal map of user activity but a glimpse into the diverse rhythms of daily life that synchronize with the Spotify experience. As we decipher these patterns, we pave the way for more targeted strategies to enhance user engagement during peak hours and capitalize on specific moments when the music resonates most profoundly with our audience.

Turning our attention to the collaborative symphony of playlists, our analysis has exposed the intricate dance of musical preferences. From the crescendo of popular playlist types to the harmonious collaboration dynamics that underpin the creation of shared playlists, we've uncovered the nuanced ways in which users curate and consume music communally. These findings illuminate opportunities to refine playlist curation algorithms, ensuring that users are not only passive consumers but active contributors to the Spotify ecosystem, fostering a sense of collective musical identity. As we reflect on the symphony of Spotify data, it becomes clear that these insights are not merely notes on a page but a roadmap guiding the platform towards a more harmonious and personalized user experience.