Data Sorting and Clustering Project

Pengxi Chen
McMaster University
July 21, 2024

[2]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
spotify_file_path = 'SpotifyFeatures.csv'
spotify_data = pd.read_csv(spotify_file_path)

# Get the dimensions of the dataset
num_observations, num_variables = spotify_data.shape
num_observations, num_variables
```

[2]: (232725, 18)

[3]:
```
# Check if 'track_id' is unique
is_track_id_unique = spotify_data['track_id'].is_unique

# Count the number of duplicated 'track_id'
num_duplicated_track_id = spotify_data['track_id'].duplicated().sum()

# Drop duplicates based on 'track_id'
spotify_data_cleaned = spotify_data.drop_duplicates(subset='track_id')

# Output results
print("Is 'track_id' unique:", is_track_id_unique)
print("Number of duplicated 'track_id':", num_duplicated_track_id)
print("Shape of the dataset after removing duplicates:", spotify_data_cleaned.
 ↪shape)
```

```
Is 'track_id' unique: False
Number of duplicated 'track_id': 55951
Shape of the dataset after removing duplicates: (176774, 18)
```

[4]:
```
# Checking the data types of the specified variables in the Spotify dataset

# List of specified variables
specified_variables = ['genre', 'artist_name', 'track_name', 'popularity',␣
 ↪'acousticness',
```

```
                      'danceability', 'duration_ms', 'energy',␣
↪'instrumentalness', 'key',
                      'liveness', 'loudness', 'mode', 'speechiness', 'tempo',
                      'time_signature', 'valence']

# Extracting data types of the specified variables
data_types = spotify_data_cleaned[specified_variables].dtypes
data_types
```

[4]:
```
genre                object
artist_name          object
track_name           object
popularity            int64
acousticness        float64
danceability        float64
duration_ms           int64
energy              float64
instrumentalness    float64
key                  object
liveness            float64
loudness            float64
mode                 object
speechiness         float64
tempo               float64
time_signature       object
valence             float64
dtype: object
```

[5]:
```
# Finding the number of different genres in the Spotify dataset

num_genres = spotify_data_cleaned['genre'].nunique()
num_genres
```

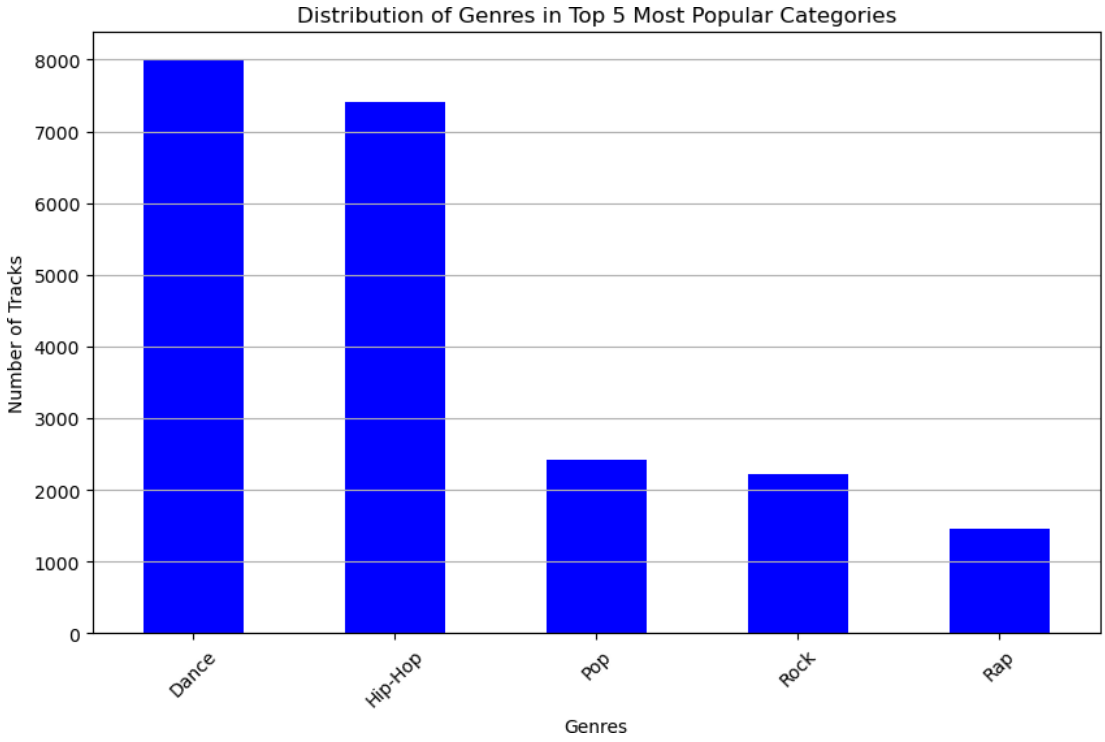
[5]:
```
27
```

[6]:
```
# Computing the average popularity of each genre

# Group by genre and compute average popularity
average_popularity_per_genre = spotify_data_cleaned.
↪groupby('genre')['popularity'].mean()

# Sorting genres by average popularity and getting the top 5 genres
top_5_genres = average_popularity_per_genre.sort_values(ascending=False).head(5)

# Selecting tracks related to the top 5 genres
top_genres_tracks = spotify_data_cleaned[spotify_data_cleaned['genre'].
↪isin(top_5_genres.index)]
```

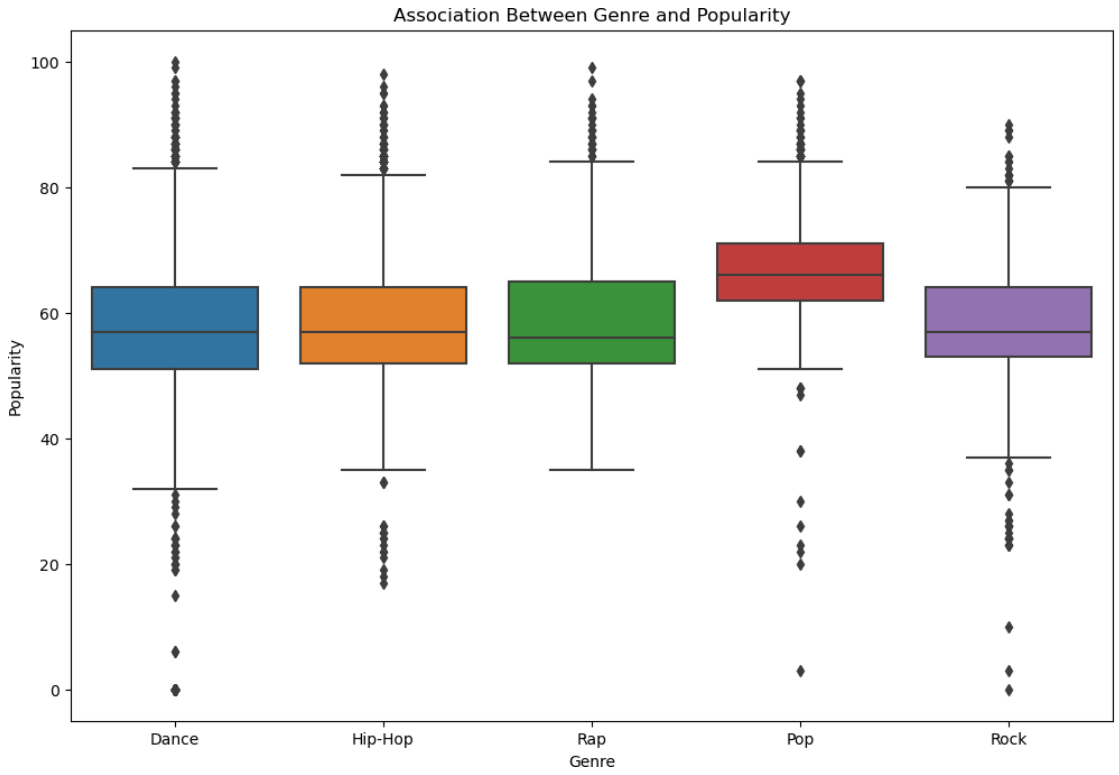
```
top_5_genres, top_genres_tracks.shape
```

[6]:
```
(genre
 Pop        67.064957
 Rap        59.515797
 Rock       58.767849
 Hip-Hop    58.516660
 Dance      57.351541
 Name: popularity, dtype: float64,
 (21495, 18))
```

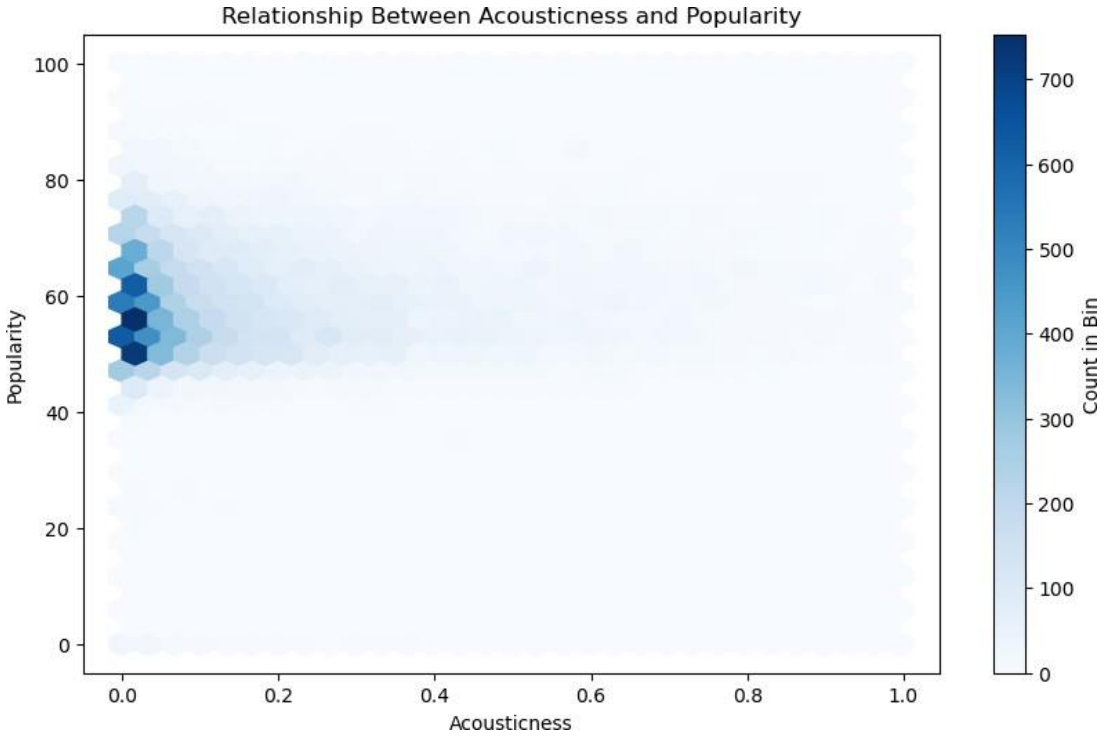
[9]:
```
# Exploring the distribution of genre in the subset of tracks related to the top 5 genres

# Count the number of tracks per genre in the subset
genre_distribution = top_genres_tracks['genre'].value_counts()

# Plotting the distribution plt.figure(figsize=(10,
6)) genre_distribution.plot(kind='bar',
color='blue')
plt.title('Distribution of Genres in Top 5 Most Popular Categories')
plt.xlabel('Genres')
plt.ylabel('Number of Tracks')
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.show()
```



[14]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Creating a standard boxplot without adjusting the widths
plt.figure(figsize=(12, 8))
sns.boxplot(x='genre', y='popularity', data=top_genres_tracks)
plt.title('Association Between Genre and Popularity')
plt.xlabel('Genre')
plt.ylabel('Popularity')
plt.show()
```



[13]:
```
# Exploring the relationship between acousticness and popularity

# Using a hexbin plot to avoid overplotting in a large dataset
plt.figure(figsize=(10, 6))
plt.hexbin(top_genres_tracks['acousticness'], top_genres_tracks['popularity'],␣
↪gridsize=30, cmap='Blues')
plt.colorbar(label='Count in Bin')
plt.xlabel('Acousticness')
plt.ylabel('Popularity')
plt.title('Relationship Between Acousticness and Popularity')
plt.show()
```


Relationship Between Acousticness and Popularity
Popularity
Acousticness
Count in Bin
0
20
40
60
80
100
0.0
0.2
0.4
0.6
0.8
1.0
0
100
200
300
400
500
600
700