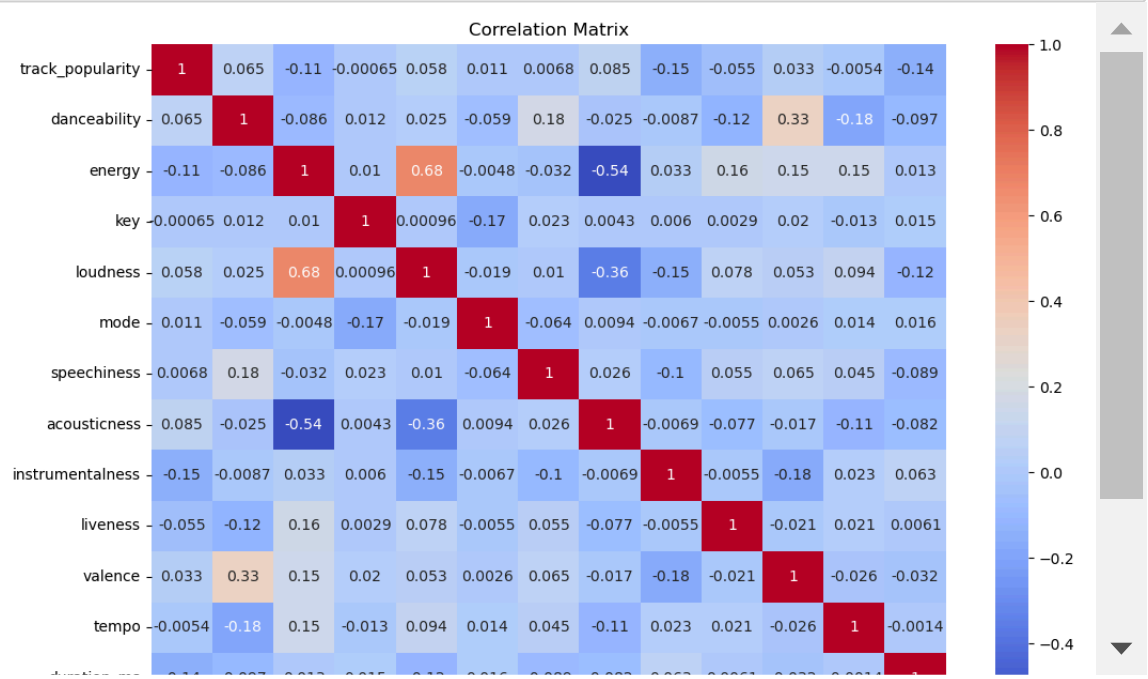
In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.neighbors import NearestNeighbors

# Enable inline plotting
%matplotlib inline
```

In [2]:
```
# Load the dataset
df = pd.read_csv('spotify dataset.csv')

# Display the first few rows of the dataset
print(df.head())

# Check for missing values
print(df.isnull().sum())

# Fill missing values with the mean of the column
imputer = SimpleImputer(strategy='mean')
df_imputed = pd.DataFrame(imputer.fit_transform(df.select_dtypes(include=[r
df_imputed.columns = df.select_dtypes(include=[np.number]).columns

# Identify non-numeric columns and apply one-hot encoding
non_numeric_columns = df.select_dtypes(include=['object']).columns
df_encoded = pd.get_dummies(df, columns=non_numeric_columns)

# Fill missing values in non-numeric columns
df_encoded = df_encoded.fillna('')

# Standardize the encoded data
scaler = StandardScaler()
df_scaled = scaler.fit_transform(df_encoded.select_dtypes(include=[np.numbe

# Convert the scaled data back to a DataFrame
df_scaled = pd.DataFrame(df_scaled, columns=df_encoded.select_dtypes(inclu
```

```
                track_id                                          track_n
ame  \
0  6f807x0ima9a1j3VPbc7VN  I Don't Care (with Justin Bieber) - Loud Luxu
r...
1  0r7CVbZTWZgbTCYdfa2P31                    Memories - Dillon Francis Re
mix
2  1z1Hg7Vb0AhHDiEmnDE79l                    All the Time - Don Diablo Re
mix
3  75FpbthrwQmzHlBJLuGdC7                  Call You Mine - Keanu Silva Re
mix
4  1e8PAfcKUYoKkxPhrHqw4x            Someone You Loved - Future Humans Re
mix

       track_artist  track_popularity          track_album_id  \
0        Ed Sheeran                66  2oCs0DGTsRO98Gh5ZSl2Cx
1          Maroon 5                67  63rPSO264uRjW1X5E6cWv6
2      Zara Larsson                70  1HoSmj2eLcsrR0vE9gThr4
3  The Chainsmokers                60  1nqYsOef1yKKuGOVchbsk6
4     Lewis Capaldi                69  7m7vv9wlQ4i0LFuJiE2zsQ

                                  track_album_name track_album_release_
date  \
0  I Don't Care (with Justin Bieber) [Loud Luxury...               2019-0
6-14
1                    Memories (Dillon Francis Remix)               2019-1
2-13
2                    All the Time (Don Diablo Remix)               2019-0
7-05
3                        Call You Mine - The Remixes               2019-0
7-19
4             Someone You Loved (Future Humans Remix)               2019-0
3-05

  playlist_name             playlist_id playlist_genre  ... key  loudness
\
0     Pop Remix  37i9dQZF1DXcZDD7cfEKhW            pop  ...   6    -2.634
1     Pop Remix  37i9dQZF1DXcZDD7cfEKhW            pop  ...  11    -4.969
2     Pop Remix  37i9dQZF1DXcZDD7cfEKhW            pop  ...   1    -3.432
3     Pop Remix  37i9dQZF1DXcZDD7cfEKhW            pop  ...   7    -3.778
4     Pop Remix  37i9dQZF1DXcZDD7cfEKhW            pop  ...   1    -4.672

   mode  speechiness  acousticness  instrumentalness  liveness  valence
\
0     1       0.0583        0.1020          0.000000    0.0653    0.518
1     1       0.0373        0.0724          0.004210    0.3570    0.693
2     0       0.0742        0.0794          0.000023    0.1100    0.613
3     1       0.1020        0.0287          0.000009    0.2040    0.277
4     1       0.0359        0.0803          0.000000    0.0833    0.725

     tempo  duration_ms
0  122.036       194754
1   99.972       162600
2  124.008       176616
3  121.956       169093
4  123.976       189052

[5 rows x 23 columns]
track_id                  0
track_name                5
track_artist              5
track_popularity          0
```

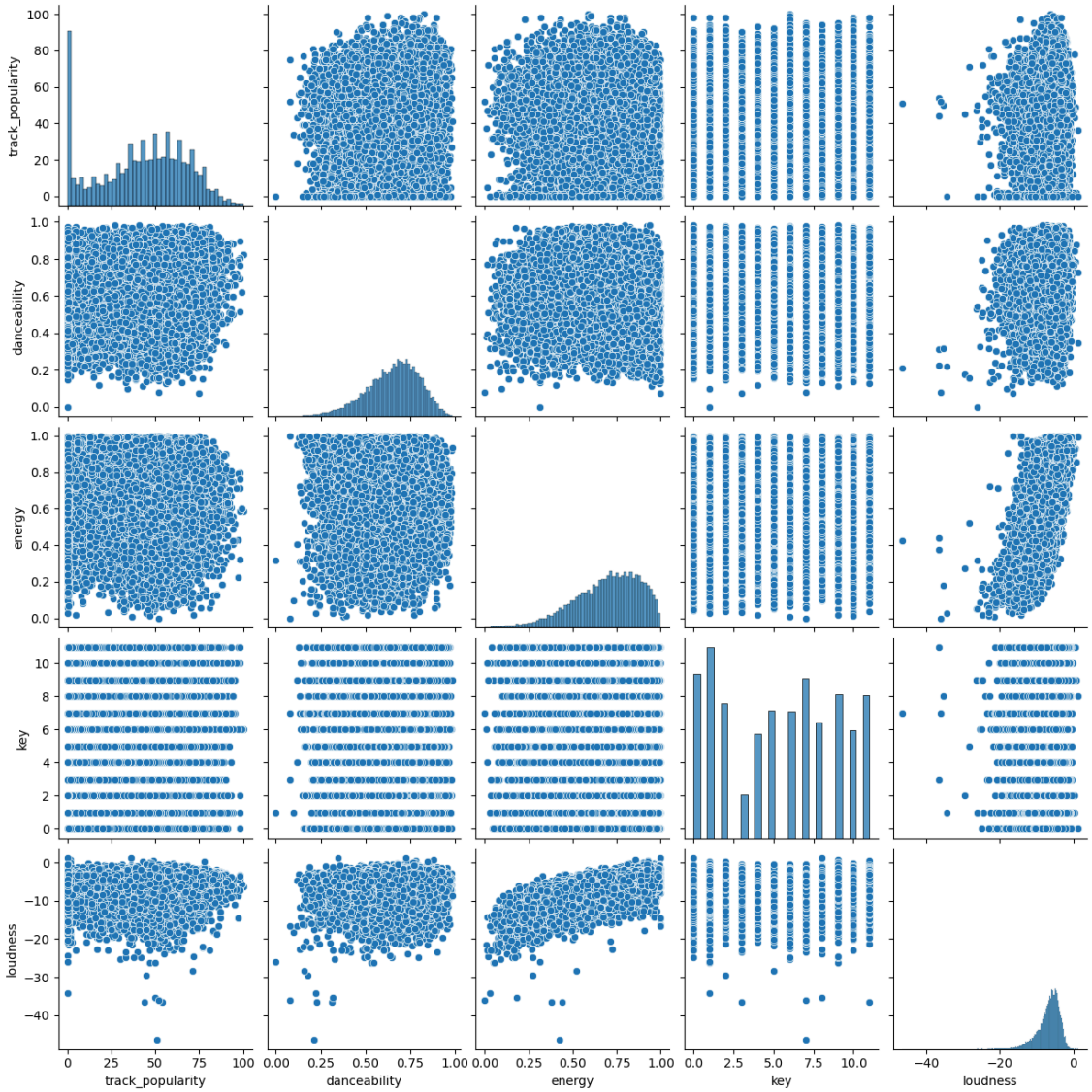
```
track_album_id              0
track_album_name            5
track_album_release_date    0
playlist_name               0
playlist_id                 0
playlist_genre              0
playlist_subgenre           0
danceability                0
energy                      0
key                         0
loudness                    0
mode                        0
speechiness                 0
acousticness                0
instrumentalness            0
liveness                    0
valence                     0
tempo                       0
duration_ms                 0
dtype: int64
```

In [7]:
```
# Select a subset of relevant features for correlation matrix
selected_features = df_scaled.columns[:20]  # Adjust the number of features
df_reduced = df_scaled[selected_features]

# Plotting the correlation matrix for the reduced data
plt.figure(figsize=(12, 8))
sns.heatmap(df_reduced.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```

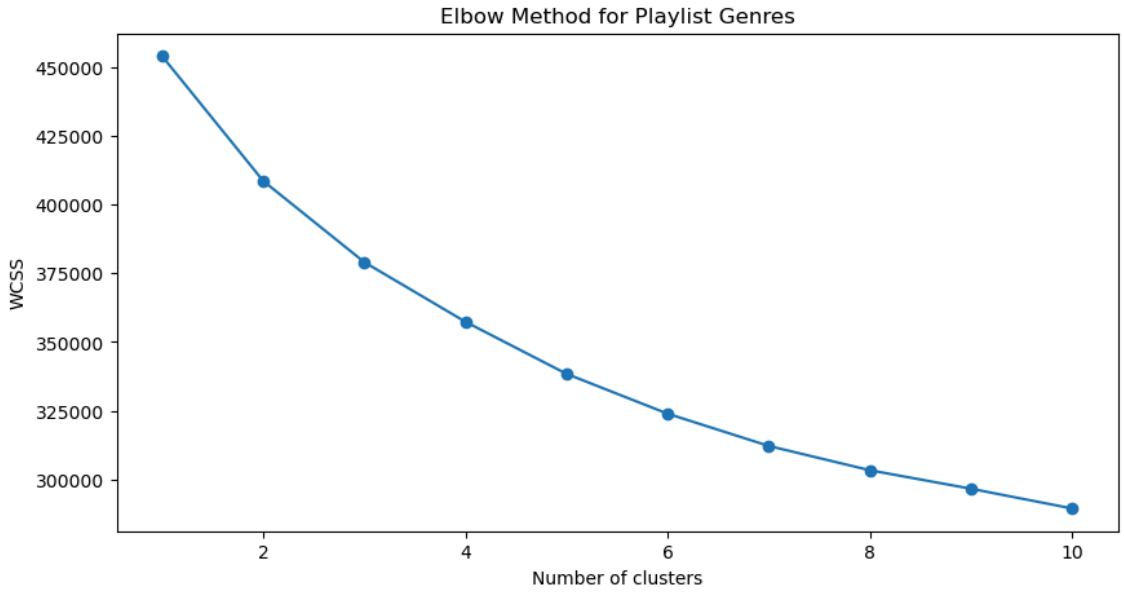


In [4]:
```
# Select a smaller subset for pairplot
pairplot_features = df_encoded.columns[:5]  # Adjust the number of features
sns.pairplot(df_encoded[pairplot_features])
plt.show()
```

```
C:\Users\DELL\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWa
rning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



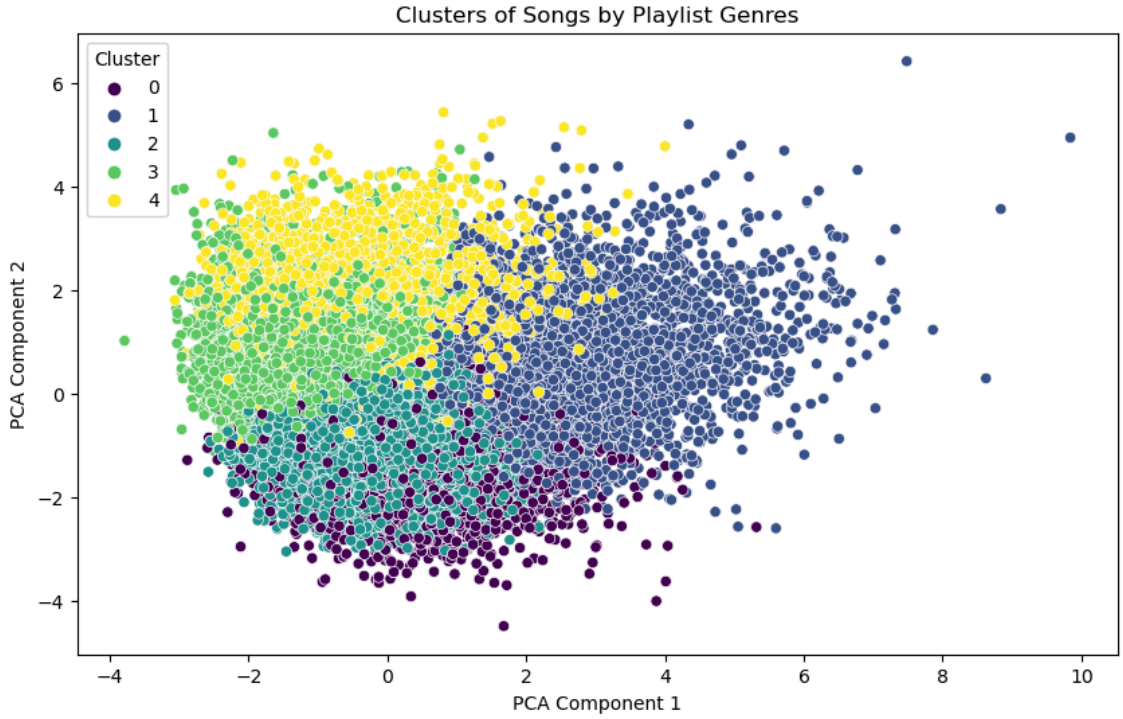
In [5]:
```
# Assuming 'playlist_genre' is a column in the original dataset
# One-hot encode playlist genres
df_genre = pd.get_dummies(df['playlist_genre'])

# Concatenate the one-hot encoded genres with the scaled features
df_genre_scaled = pd.concat([df_scaled, df_genre], axis=1)

# Fill missing values in the genre data
df_genre_scaled = df_genre_scaled.fillna(0)

# Finding the optimal number of clusters using the elbow method for genres
wcss_genre = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10
    kmeans.fit(df_genre_scaled)
    wcss_genre.append(kmeans.inertia_)

# Plotting the elbow graph for genres
plt.figure(figsize=(10, 5))
plt.plot(range(1, 11), wcss_genre, marker='o')
plt.title('Elbow Method for Playlist Genres')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()

# Fitting KMeans to the genre data
kmeans_genre = KMeans(n_clusters=5, init='k-means++', max_iter=300, n_init=
y_kmeans_genre = kmeans_genre.fit_predict(df_genre_scaled)

# Adding cluster labels to the original dataframe
df['Cluster_Genre'] = y_kmeans_genre
```



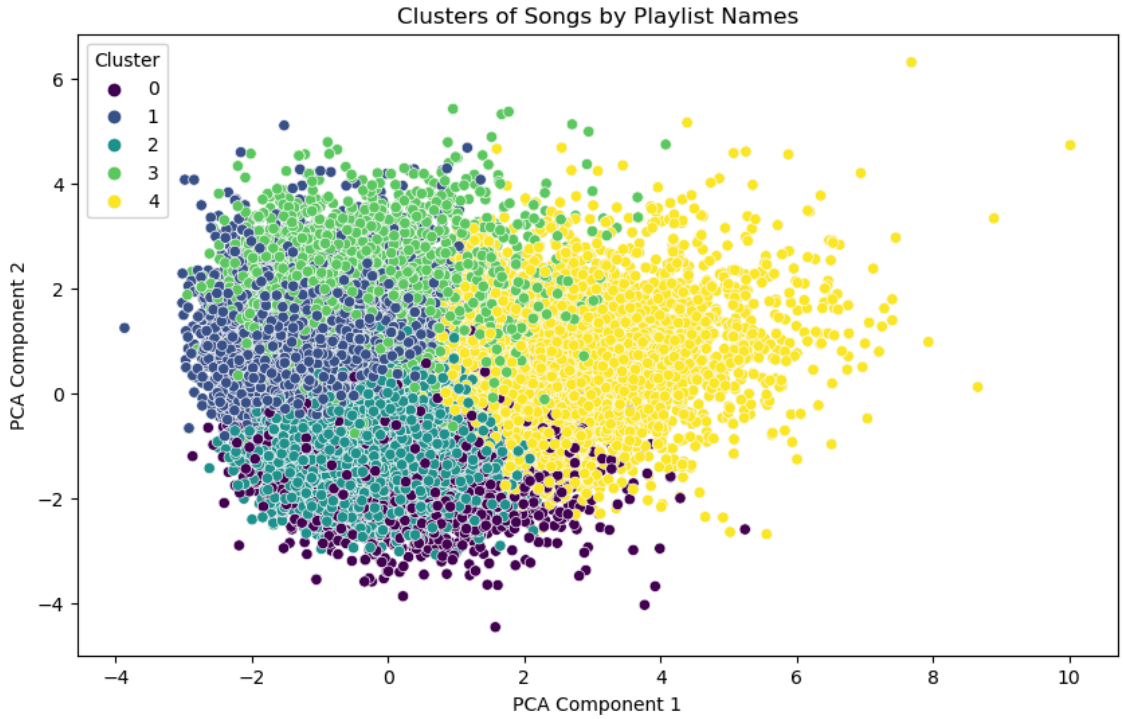
In [6]:
```
# Check if 'playlist_name' is a column in the original dataset
if 'playlist_name' not in df.columns:
    raise ValueError("Column 'playlist_name' is not found in the dataset."

# One-hot encode playlist names
df_name = pd.get_dummies(df['playlist_name'])

# Concatenate the one-hot encoded names with the scaled features
df_name_scaled = pd.concat([df_scaled, df_name], axis=1)

# Fill missing values in the name data
df_name_scaled = df_name_scaled.fillna(0)

# Print the shape and first few rows of df_name_scaled for debugging
print(df_name_scaled.shape)
print(df_name_scaled.head())

# Finding the optimal number of clusters using the elbow method for names
wcss_name = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=1
    kmeans.fit(df_name_scaled)
    wcss_name.append(kmeans.inertia_)

# Plotting the elbow graph for names
plt.figure(figsize=(10, 5))
plt.plot(range(1, 11), wcss_name, marker='o')
plt.title('Elbow Method for Playlist Names')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()

# Fitting KMeans to the name data
kmeans_name = KMeans(n_clusters=5, init='k-means++', max_iter=300, n_init=1
y_kmeans_name = kmeans_name.fit_predict(df_name_scaled)

# Adding cluster labels to the original dataframe
df['Cluster_Name'] = y_kmeans_name

# Print the first few rows of the dataframe with the cluster labels for del
print(df.head())
```

```
(32833, 462)
   track_popularity  danceability    energy       key  loudness      m
ode  \
0          0.941531      0.642049  1.201614  0.173200  1.367123  0.876
177
1          0.981557      0.490412  0.643317  1.557627  0.585766  0.876
177
2          1.101635      0.138889  1.284529 -1.211227  1.100090 -1.141
322
3          0.701374      0.435271  1.279002  0.450085  0.984309  0.876
177
4          1.061609     -0.033426  0.742815 -1.211227  0.685151  0.876
177

   speechiness  acousticness  instrumentalness  liveness  ...  ¡Viva L
atino!  \
0    -0.481362     -0.333898         -0.377953 -0.809230  ...
False
1    -0.688642     -0.468670         -0.359177  1.081061  ...
```

In [8]:
```
# Visualizing genre clusters using PCA
pca_genre = PCA(n_components=2)
pca_data_genre = pca_genre.fit_transform(df_genre_scaled)

# Convert PCA data to DataFrame for plotting
pca_df_genre = pd.DataFrame(pca_data_genre, columns=['PCA1', 'PCA2'])
pca_df_genre['Cluster'] = y_kmeans_genre

# Scatter plot for genre clusters
plt.figure(figsize=(10, 6))
sns.scatterplot(x='PCA1', y='PCA2', hue='Cluster', palette='viridis', data=
plt.title('Clusters of Songs by Playlist Genres')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.show()
```



In [9]:
```
# Visualizing name clusters using PCA
pca_name = PCA(n_components=2)
pca_data_name = pca_name.fit_transform(df_name_scaled)

# Convert PCA data to DataFrame for plotting
pca_df_name = pd.DataFrame(pca_data_name, columns=['PCA1', 'PCA2'])
pca_df_name['Cluster'] = y_kmeans_name

# Scatter plot for name clusters
plt.figure(figsize=(10, 6))
sns.scatterplot(x='PCA1', y='PCA2', hue='Cluster', palette='viridis', data=
plt.title('Clusters of Songs by Playlist Names')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.show()
```



In [11]:
```
# Define a function to get song recommendations
def get_recommendations(song_index, model, df_scaled, df):
    distances, indices = model.kneighbors([df_scaled.iloc[song_index]])
    recommendations = df.iloc[indices[0]].drop(song_index)
    return recommendations

# Example usage
song_index = 0  # Index of the song you want recommendations for
recommendations = get_recommendations(song_index, model, df_scaled, df)
print("Recommendations:")
print(recommendations)
```

```
Recommendations:
                    track_id  \
29684  6f807x0ima9a1j3VPbc7VN
2371   5HOpkTTVcmZHnthgyxrIL8
1040   0ahbzg4GCq3wJzUM3cjS3N
231    0GRoERSBBky3YgdKW2w2Vc
1385   0GRoERSBBky3YgdKW2w2Vc
28680  3m0BCyPaS5BN5iQgtdPr0u
26871  3m0BCyPaS5BN5iQgtdPr0u
19886  3EFzexdYFIzbM2dusddMVk
365    3kGfazcbVvVkuZunzlLgTD

                                              track_name   track_artist
\
29684  I Don't Care (with Justin Bieber) - Loud Luxur...     Ed Sheeran
2371                   The Fox (What Does the Fox Say?)          Ylvis
1040                                          Violeta         IZ*ONE
231                                             Brave     Don Diablo
1385                                            Brave     Don Diablo
28680               Perfect (feat. Haris) - LUM!X Remix  Lucas & Steve
26871               Perfect (feat. Haris) - LUM!X Remix  Lucas & Steve
19886                                  Muñequita Linda     Juan Magán
365                                 Came Here for Love          Sigala

       track_popularity          track_album_id  \
29684                66  2oCs0DGTsRO98Gh5ZSl2Cx
2371                 65  77QwsMRvonZJn7adV47V78
1040                 67  2UBE2MgNdsGa9OCSbvwdEQ
231                  68  1vlt3ZZeHbtRWKOFu45TEJ
1385                 68  1vlt3ZZeHbtRWKOFu45TEJ
28680                59  45DODwRRHNzuqPq18lGAWl
26871                59  45DODwRRHNzuqPq18lGAWl
19886                61  2ndIHUzRvnc6rLoSxFVfd3
365                  68  22x1g0NEicPMxuEOXlGUw5

                                       track_album_name  \
29684  I Don't Care (with Justin Bieber) [Loud Luxury...
2371                   The Fox (What Does The Fox Say?)
1040                                           HEART*IZ
231                                               Brave
1385                                              Brave
28680               Perfect (feat. Haris) [LUM!X Remix]
26871               Perfect (feat. Haris) [LUM!X Remix]
19886                                  Muñequita Linda
365                                 Came Here for Love

      track_album_release_date  \
29684               2019-06-14
2371                2013-09-02
1040                2019-04-01
231                 2019-04-26
1385                2019-04-26
28680               2019-11-29
26871               2019-11-29
19886               2018-12-14
365                 2017-06-09

                                        playlist_name  \
29684                                 Pop EDM Remixes
2371                                    post teen pop
1040                    Best of 2019 Dance Pop: Japan
```

```
231                                          Cardio
1385   Pop - Pop UK - 2019 - Canadian Pop - 2019 - Pop
28680              Big Room EDM - by Spinnin' Records
26871                        Electro House Top Tracks
19886                                  latin hip hop
365                                 Pop Warmup 130 BPM

                 playlist_id playlist_genre  ... mode  speechiness  \
29684  4aUEH3uhbofktrFkXOOaKj            edm  ...    1       0.0583
2371   6rjxP7GQKoqqgoakzxl3PY            pop  ...    1       0.0453
1040   37i9dQZF1DXdOtZGKonFlM            pop  ...    1       0.0685
231    37i9dQZF1DWSJHnPb1f0X3            pop  ...    1       0.0543
1385   46Cl6dmeiylK6TRGXr7hHe            pop  ...    1       0.0543
28680  7xWdFCrU5Gka6qp1ODrSdK            edm  ...    1       0.0354
26871  1G0q0NK7g3C0XerNqq7GbL            edm  ...    1       0.0354
19886  3nH8aytdqNeRbcRCg3dw9q          latin  ...    1       0.0465
365    37i9dQZF1DX3PIAZMcbo2T            pop  ...    1       0.0431

       acousticness  instrumentalness  liveness  valence    tempo  \
29684       0.10200          0.000000    0.0653    0.518  122.036
2371        0.10700          0.000000    0.1190    0.546  128.008
1040        0.00373          0.000000    0.1040    0.669  115.012
231         0.16900          0.000002    0.1340    0.542  119.850
1385        0.16900          0.000002    0.1340    0.542  119.850
28680       0.00266          0.000003    0.0581    0.509  127.932
26871       0.00266          0.000003    0.0581    0.509  127.932
19886       0.00402          0.000993    0.0946    0.415  130.054
365         0.03130          0.000000    0.1220    0.720  124.994

       duration_ms  Cluster_Genre  Cluster_Name
29684       194754              2             2
2371        213708              2             2
1040        200913              2             2
231         184027              2             2
1385        184027              2             2
28680       187074              2             2
26871       187074              2             2
19886       210524              2             2
365         202999              2             2

[9 rows x 25 columns]

C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarnin
g: X does not have valid feature names, but NearestNeighbors was fitted w
ith feature names
  warnings.warn(
```

In [ ]: