Exploratory Data Analysis (EDA) on Spotify Music Dataset

Spotify is a leading digital music streaming platform that provides users with access to over 70 million songs, podcasts, and other audio content from artists, creators, and record labels worldwide. Founded in 2006 and launched in 2008, Spotify has revolutionized how people consume music by offering both free (ad-supported) and premium (ad-free, offline listening) services, making it one of the most widely used music streaming services globally.

Importing Libraries:

In [4]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import warnings
warnings.filterwarnings('ignore')
```

Project Overview:

This Analysis aims to uncover insights into how people interact with music on spotify. we will examine:

- How music trends have evolved over the years.
- What audio features correlate with the song popularity.
- How user prefernces vary by genere,artist and the time period.

Purpose of the Analysis:

The aim of this project is to analyze a large dataset of Spotify songs to uncover key trends, patterns, and insights about music releases, popularity, genres, and artist performance over time. The goal is to provide actionable business insights that help understand what factors contribute to a song's popularity and how music trends are evolving year by year.

Loading Dataset:

In [10]:
```
df=pd.read_csv("Downloads/data.csv")
```

In [11]:
```
df
```

Out[11]:

	valence	year	acousticness	artists	danceability	duration_ms
0	0.0594	1921	0.98200	['Sergei Rachmaninoff', 'James Levine', 'Berli...	0.279	831667
1	0.9630	1921	0.73200	['Dennis Day']	0.819	180533
2	0.0394	1921	0.96100	['KHP Kridhamardawa Karaton Ngayogyakarta Hadi...	0.328	500062
3	0.1650	1921	0.96700	['Frank Parker']	0.275	210000
4	0.2530	1921	0.95700	['Phil Regan']	0.418	166693
...	...	...	...	...	...	...
170648	0.6080	2020	0.08460	['Anuel AA', 'Daddy Yankee', 'KAROL G', 'Ozuna...	0.786	301714
170649	0.7340	2020	0.20600	['Ashnikko']	0.717	150654
170650	0.6370	2020	0.10100	['MAMAMOO']	0.634	211280
170651	0.1950	2020	0.00998	['Eminem']	0.671	337147
170652	0.6420	2020	0.13200	['KEVVO', 'J Balvin']	0.856	189507

170653 rows × 19 columns

Data Inspection and Cleaning:

In [13]:
```
df.shape
```

Out[13]:
```
(170653, 19)
```

In [14]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 170653 entries, 0 to 170652
Data columns (total 19 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   valence           170653 non-null  float64
 1   year              170653 non-null  int64  
 2   acousticness      170653 non-null  float64
 3   artists           170653 non-null  object 
 4   danceability      170653 non-null  float64
 5   duration_ms       170653 non-null  int64  
 6   energy            170653 non-null  float64
 7   explicit          170653 non-null  int64  
 8   id                170653 non-null  object 
 9   instrumentalness  170653 non-null  float64
 10  key               170653 non-null  int64  
 11  liveness          170653 non-null  float64
 12  loudness          170653 non-null  float64
 13  mode              170653 non-null  int64  
 14  name              170653 non-null  object 
 15  popularity        170653 non-null  int64  
 16  release_date      170653 non-null  object 
 17  speechiness       170653 non-null  float64
 18  tempo             170653 non-null  float64
dtypes: float64(9), int64(6), object(4)
memory usage: 24.7+ MB
```

In [15]:
```
df.isnull().sum()
```

Out[15]:
```
valence             0
year                0
acousticness        0
artists             0
danceability        0
duration_ms         0
energy              0
explicit            0
id                  0
instrumentalness    0
key                 0
liveness            0
loudness            0
mode                0
name                0
popularity          0
release_date        0
speechiness         0
tempo               0
dtype: int64
```

In [16]:
```
df.columns
```

Out[16]:
```
Index(['valence', 'year', 'acousticness', 'artists', 'danceability',
       'duration_ms', 'energy', 'explicit', 'id', 'instrumentalness', 'key',
       'liveness', 'loudness', 'mode', 'name', 'popularity', 'release_date',
       'speechiness', 'tempo'],
      dtype='object')
```

In [17]:
```
df.columns.isnull().sum()
```

Out[17]:
```
0
```

In [18]:
```
df.head()
```

Out[18]:

	valence	year	acousticness	artists	danceability	duration_ms	energ
0	0.0594	1921	0.982	['Sergei Rachmaninoff', 'James Levine', 'Berli...	0.279	831667	0.21
1	0.9630	1921	0.732	['Dennis Day']	0.819	180533	0.34
2	0.0394	1921	0.961	['KHP Kridhamardawa Karaton Ngayogyakarta Hadi...	0.328	500062	0.16
3	0.1650	1921	0.967	['Frank Parker']	0.275	210000	0.30
4	0.2530	1921	0.957	['Phil Regan']	0.418	166693	0.19

In [19]:

```
df.tail()
```

Out[19]:

	valence	year	acousticness	artists	danceability	duration_ms	e
170648	0.608	2020	0.08460	['Anuel AA', 'Daddy Yankee', 'KAROL G', 'Ozuna...	0.786	301714	
170649	0.734	2020	0.20600	['Ashnikko']	0.717	150654	
170650	0.637	2020	0.10100	['MAMAMOO']	0.634	211280	
170651	0.195	2020	0.00998	['Eminem']	0.671	337147	
170652	0.642	2020	0.13200	['KEVVO', 'J Balvin']	0.856	189507	

In [20]:

```
df.describe()
```

Out[20]:

	valence	year	acousticness	danceability	duration_m
count	170653.000000	170653.000000	170653.000000	170653.000000	1.706530e+0
mean	0.528587	1976.787241	0.502115	0.537396	2.309483e+0
std	0.263171	25.917853	0.376032	0.176138	1.261184e+0
min	0.000000	1921.000000	0.000000	0.000000	5.108000e+0
25%	0.317000	1956.000000	0.102000	0.415000	1.698270e+0
50%	0.540000	1977.000000	0.516000	0.548000	2.074670e+0
75%	0.747000	1999.000000	0.893000	0.668000	2.624000e+0
max	1.000000	2020.000000	0.996000	0.988000	5.403500e+0

In [21]:

```
duplicate_rows = df.duplicated().sum()
```

In [22]:

```
duplicate_rows
```

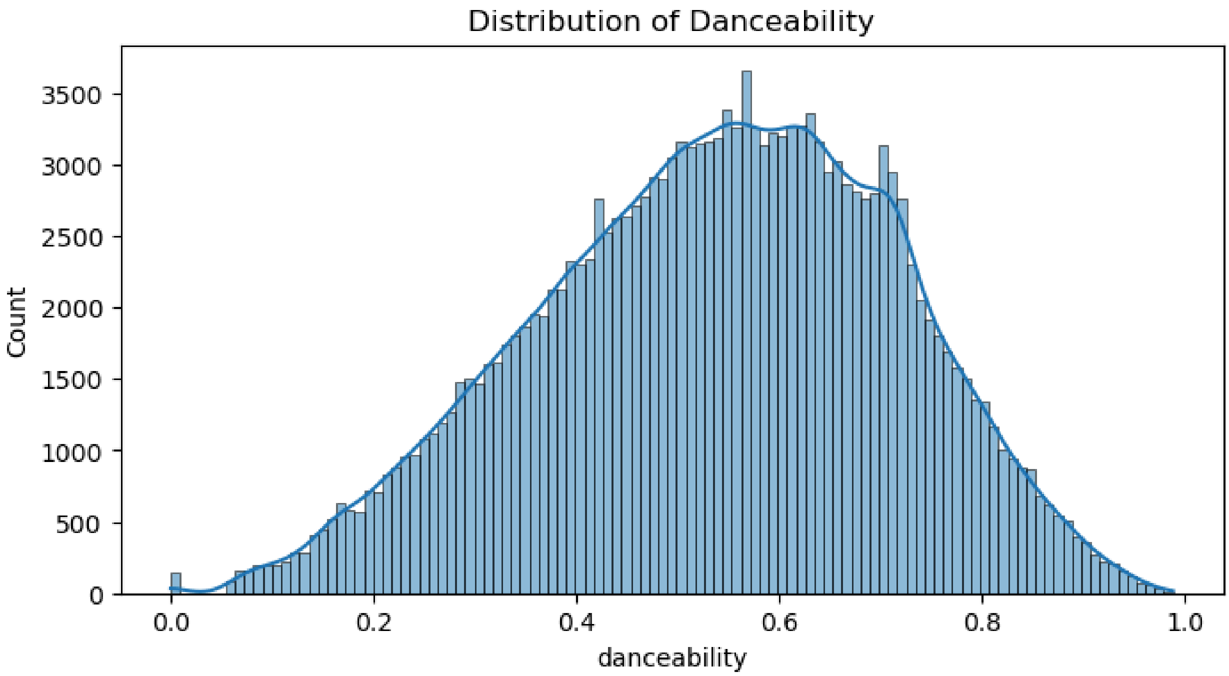
Out[22]: 0

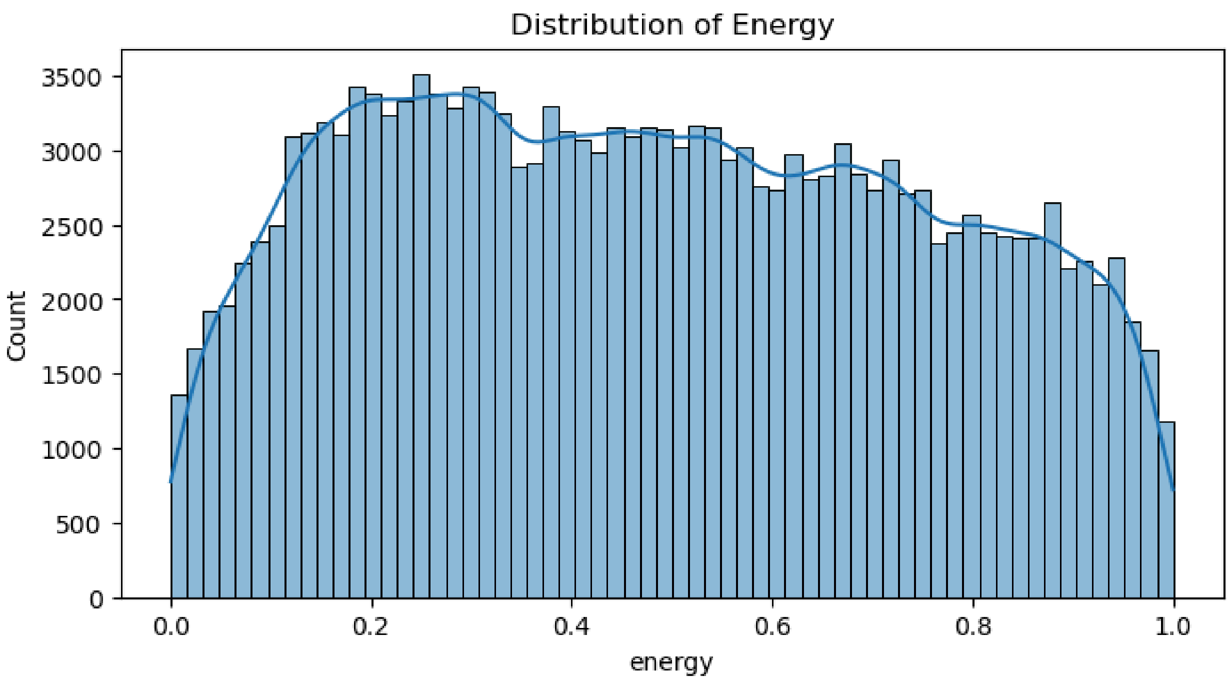
Exploratory Data Analysis(EDA):

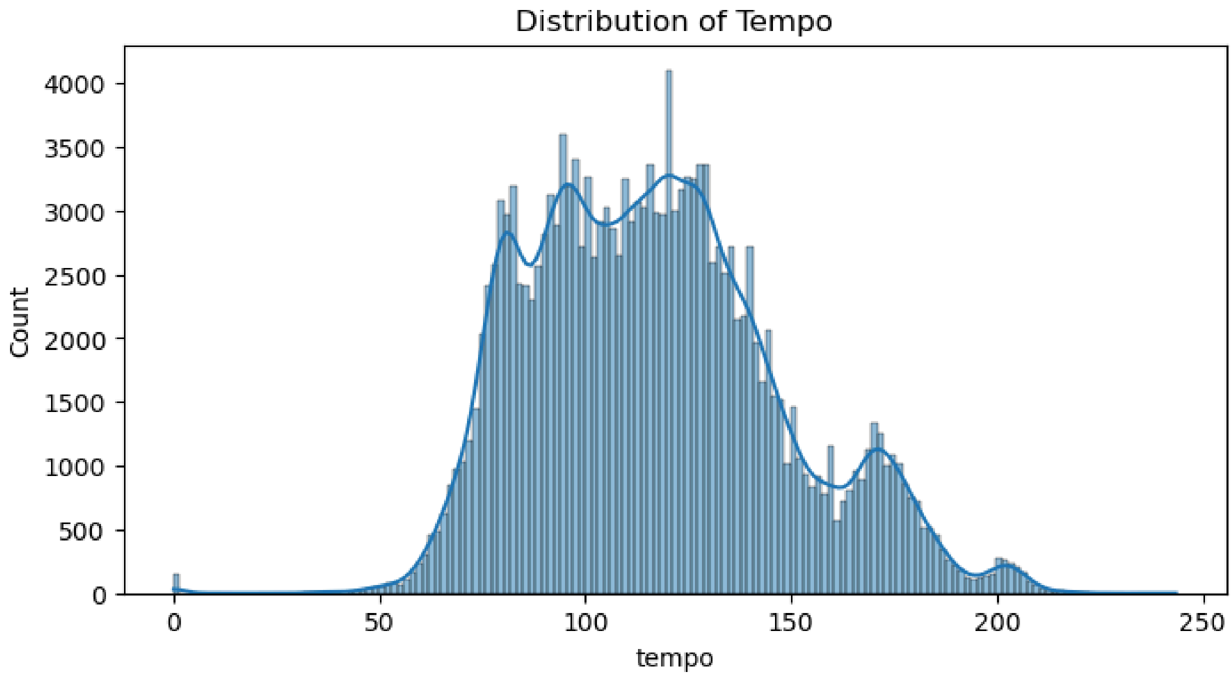
Feature Distributions:

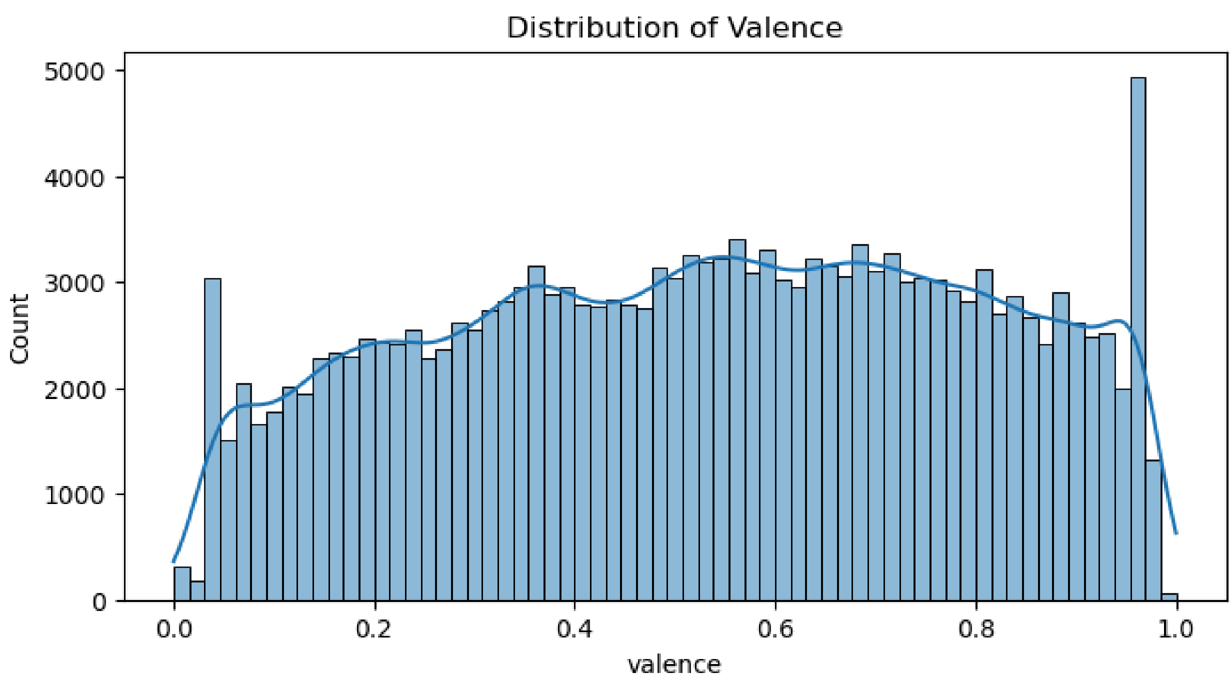
In [25]:

```
Features=['danceability','energy','tempo','valence']
for feature in Features:
    plt.figure(figsize=(8,4))
    sns.histplot(df[feature],kde=True)
    plt.title(f'Distribution of {feature.capitalize()}')
    plt.show()
```


Distribution of Danceability
Count
3500
3000
2500
2000
1500
1000
500
0
0.0
0.2
0.4
0.6
0.8
1.0
danceability


Distribution of Energy
Count
3500
3000
2500
2000
1500
1000
500
0
0.0
0.2
0.4
0.6
0.8
1.0
energy



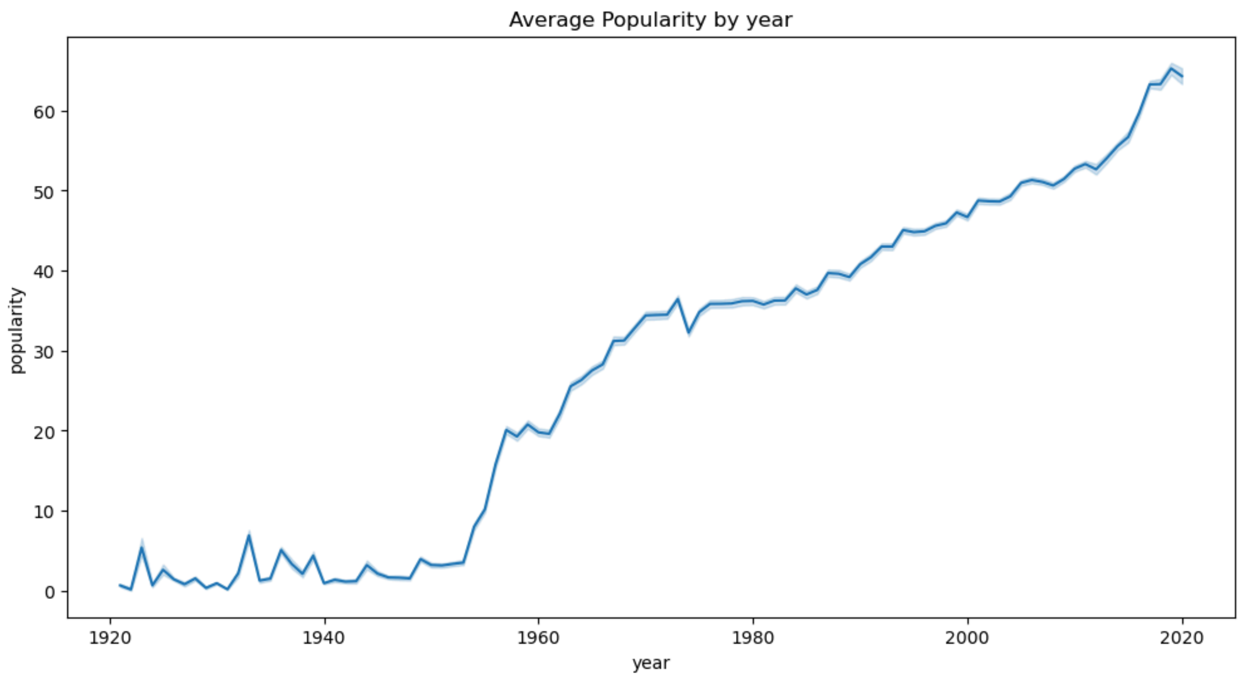


Key Insights:

1. Danceablity shows a normal distribution centered around 0.55
2. Energy is slightly left-skewed with most tracks having moderate energy.
3. Tempo has a bimodal distribution.
4. Valence(muscial postiveness) is a relatively evenly distributed.

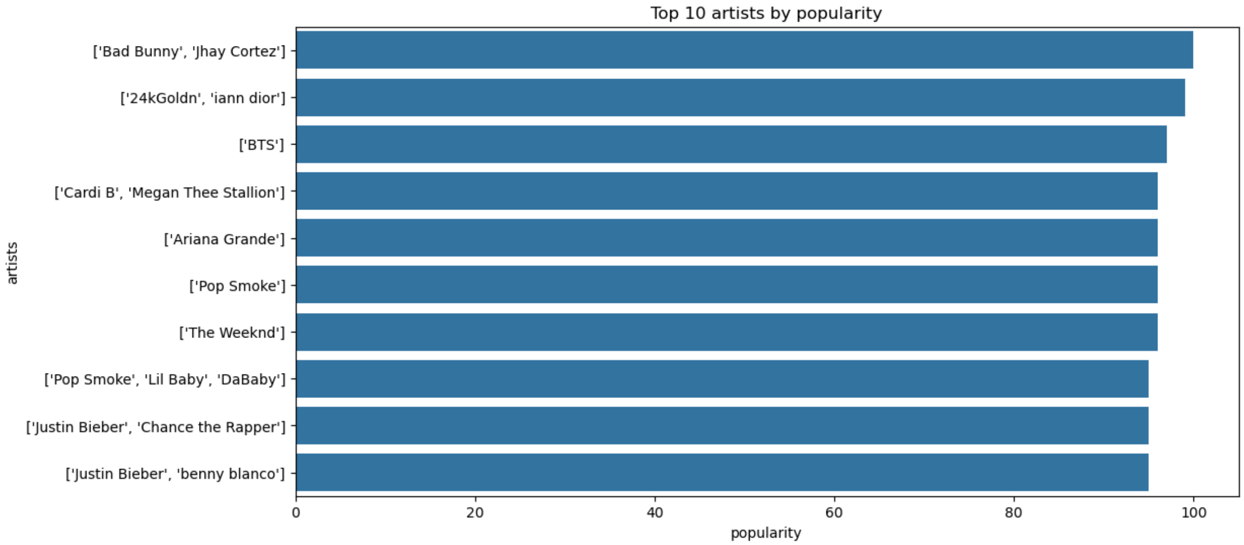
Popularity Trends over the time:

In [29]:
```
plt.figure(figsize=(12,6))
sns.lineplot(x='year',y='popularity',data=df)
plt.title('Average Popularity by year')
plt.show()
```



Examine popularity by genre revealed.

In [31]:
```
top_artists = df.sort_values('popularity', ascending=False).head(10)
plt.figure(figsize=(12,6))
sns.barplot(data=top_artists, x='popularity', y='artists')
plt.title('Top 10 artists by popularity')
plt.show()
```



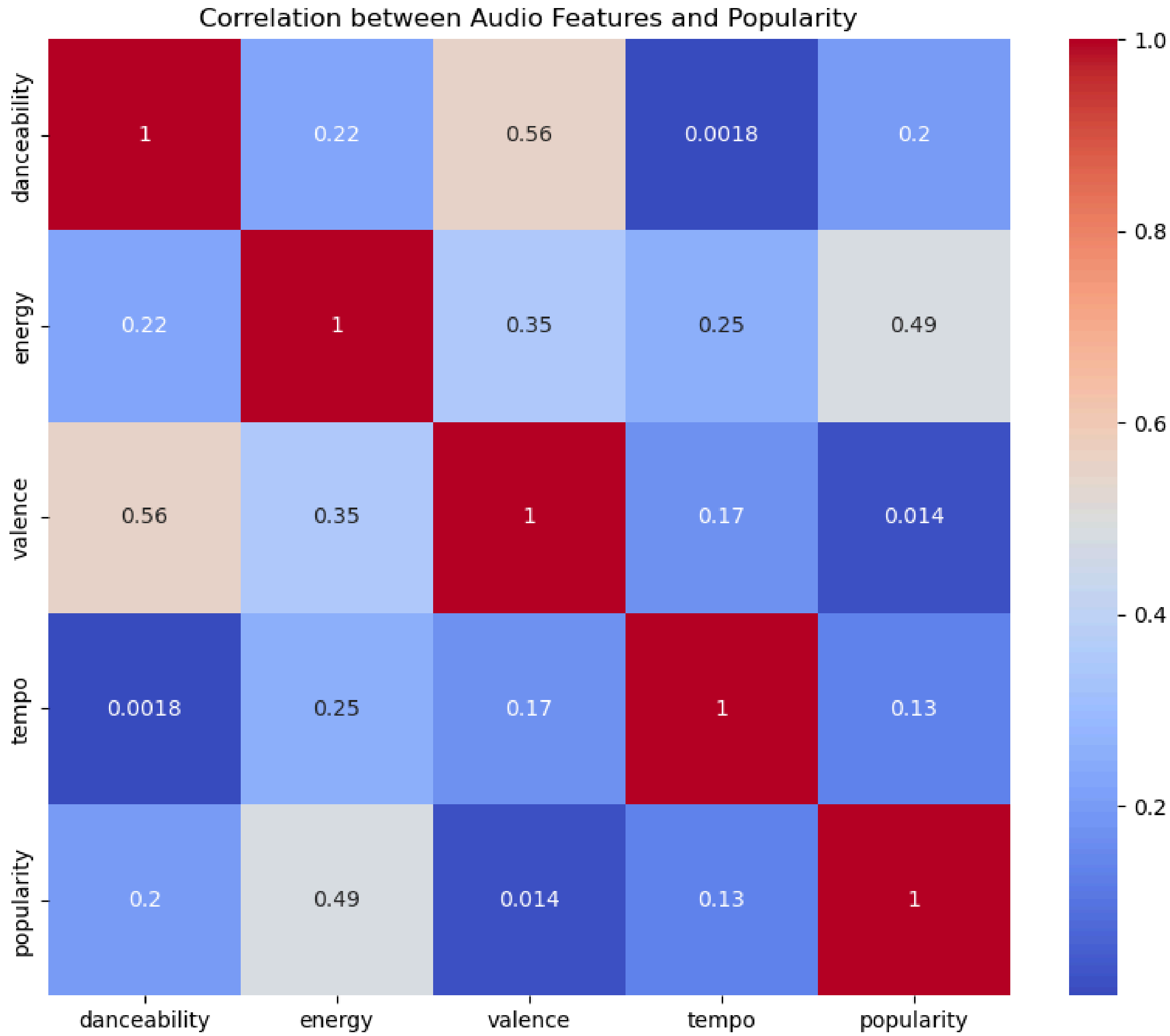
Keyinsights:

1. Artists 'jhay cortez', 'iann dior', 'BTS' are the consistently popular genre.
2. Some niche genres show high popluarity within thier segments.
3. Genre Popularity correlates with mainstream appeal.

Audio Features vs. Popularity

In [35]:

```
corr_matrix= df[['danceability','energy','valence','tempo','popularity']].corr
plt.figure(figsize=(10,8))
sns.heatmap(corr_matrix,annot=True,cmap='coolwarm')
plt.title('Correlation between Audio Features and Popularity')
plt.show()
```



Keyinsights:

1. Interesting correlations.
2. Danceability shows moderate postive correlation with the popularity
3. Energy has a weaker but still postive relationship.
4. Valence shows the minimal direct correlation.
5. Tempo has almost no correlation with the popularity.

Recommendations for Artists and Industry

Based on our analysis, we recommend:

- For Artists:

1. Focus on the danceability in track production.
2. Maintain moderate to high engery levels.
3. Consider pop or dance pop genre for mainstream appeal.
4. Experiment with the tempo as it shows wide variation in popular tracks.

- For Spotify:

1. Highlights danceable tracks in algorithmic recommendations.
2. Consider energy levels when curating workout or focus playlists.
3. Explore niche genres that show unexpected popularity.

- For Listeners:

1. Explore beyond just popular tracks- many great songs exist across all popularity levels.
2. Use audio features to discover new music matching your prefernces.

Future Work:

Potential extensions for this analysis

- Incoperate lyrics analysis for deeper insights.
- Examine geographical trends in music prefernces.
- Build predictive models for song popularity.
- Analyze playlist compostions patterns.
- Study the impact of collaborations on track resources.

Conclusion

Our Comprehensive analysis of spotify data revealed fascinating insights into what makes music popular. key takeaways include:

1. Danceability and energy are important but not sole determinants of popularity.
2. Genre plays a significant role in a track sucess.
3. Popularity has a generally increased over time.
4. There's more to music than just popularity- many great tracks exist

across all levels.

This Analysis provides valueable insights for artists,music industry professionals and listeners alike to better understand and navigate the evolving music landscape.

In [ ]:
```

```