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

import warnings
warnings.filterwarnings("ignore")
```

DATA CLEANING

In [2]:
```
df=pd.read_csv("Desktop/musicgenre.csv")
df
```

Out[2]:

	Artist Name	Track Name	Popularity	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness
0	Bruno Mars	That's What I Like (feat. Gucci Mane)	60.0	0.854	0.564	1.0	-4.964	1	0.0485	0.017100	NaN	0.0849
1	Boston	Hitch a Ride	54.0	0.382	0.814	3.0	-7.230	1	0.0406	0.001100	0.004010	0.1010
2	The Raincoats	No Side to Fall In	35.0	0.434	0.614	6.0	-8.334	1	0.0525	0.486000	0.000196	0.3940
3	Deno	Lingo (feat. J.I & Chunkz)	66.0	0.853	0.597	10.0	-6.528	0	0.0555	0.021200	NaN	0.1220
4	Red Hot Chili Peppers	Nobody Weird Like Me - Remastered	53.0	0.167	0.975	2.0	-4.279	1	0.2160	0.000169	0.016100	0.1720
...	...	...	...	...	...	...	...	...	...	...	...	...
17991	Green-House	Find Home	35.0	0.166	0.109	7.0	-17.100	0	0.0413	0.993000	0.824000	0.0984
17992	Micatone	All Gone	27.0	0.638	0.223	11.0	-10.174	0	0.0329	0.858000	0.000016	0.0705
17993	Smash Hit Combo	Peine perdue	34.0	0.558	0.981	4.0	-4.683	0	0.0712	0.000030	0.000136	0.6660
17994	Beherit	Salomon's Gate	29.0	0.215	0.805	6.0	-12.757	0	0.1340	0.001290	0.916000	0.2560
17995	The Raconteurs	Broken Boy Soldier	43.0	0.400	0.853	4.0	-5.320	0	0.0591	0.006040	0.212000	0.3340

17996 rows × 17 columns

In [3]:
```
df.shape
```

Out[3]:
```
(17996, 17)
```

In [4]:
```
df.describe()
```

Out[4]:

	Popularity	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	
count	17568.000000	17996.000000	17996.000000	15982.000000	17996.000000	17996.000000	17996.000000	17996.000000	13619.000000	179
mean	44.512124	0.543433	0.662777	5.952447	-7.910660	0.636753	0.079707	0.247082	0.177562	
std	17.426928	0.166268	0.235373	3.196854	4.049151	0.480949	0.083576	0.310632	0.304048	
min	1.000000	0.059600	0.000020	1.000000	-39.952000	0.000000	0.022500	0.000000	0.000001	
25%	33.000000	0.432000	0.509000	3.000000	-9.538000	0.000000	0.034800	0.004300	0.000089	
50%	44.000000	0.545000	0.700000	6.000000	-7.016000	1.000000	0.047400	0.081400	0.003910	
75%	56.000000	0.659000	0.860000	9.000000	-5.189000	1.000000	0.083000	0.434000	0.200000	
max	100.000000	0.989000	1.000000	11.000000	1.355000	1.000000	0.955000	0.996000	0.996000	

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

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 17996 entries, 0 to 17995
Data columns (total 17 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Artist Name         17996 non-null  object 
 1   Track Name          17996 non-null  object 
 2   Popularity          17568 non-null  float64
 3   danceability        17996 non-null  float64
 4   energy              17996 non-null  float64
 5   key                 15982 non-null  float64
 6   loudness            17996 non-null  float64
 7   mode                17996 non-null  int64  
 8   speechiness         17996 non-null  float64
 9   acousticness        17996 non-null  float64
 10  instrumentalness    13619 non-null  float64
 11  liveness            17996 non-null  float64
 12  valence             17996 non-null  float64
 13  tempo               17996 non-null  float64
 14  duration_in min/ms  17996 non-null  float64
 15  time_signature      17996 non-null  int64  
 16  Class               17996 non-null  int64  
dtypes: float64(12), int64(3), object(2)
memory usage: 2.3+ MB
```

In [6]:
```
df['Artist Name'].unique()
```

Out[6]:
```
array(['Bruno Mars', 'Boston', 'The Raincoats', ..., 'Micatone',
       'Smash Hit Combo', 'Beherit'], dtype=object)
```

In [7]:
```
len(df['Artist Name'].unique())
```

Out[7]:
```
9149
```

In [8]:
```
df['Track Name'].unique()
```

Out[8]:
```
array(["That's What I Like (feat. Gucci Mane)", 'Hitch a Ride',
       'No Side to Fall In', ..., 'Peine perdue', "Salomon's Gate",
       'Broken Boy Soldier'], dtype=object)
```

In [9]:
```
len(df['Track Name'].unique())
```

Out[9]:
```
15129
```

In [10]:
```
nullPercentages=pd.DataFrame(df.isnull().sum(),columns=['Number of Null Values'])
nullPercentages['Fraction of Null Percentages']=np.round(df.isnull().sum()/df.shape[0],2)*100
nullPercentages
```

Out[10]:

	Number of Null Values	Fraction of Null Percentages
Artist Name	0	0.0
Track Name	0	0.0
Popularity	428	2.0
danceability	0	0.0
energy	0	0.0
key	2014	11.0
loudness	0	0.0
mode	0	0.0
speechiness	0	0.0
acousticness	0	0.0
instrumentalness	4377	24.0
liveness	0	0.0
valence	0	0.0
tempo	0	0.0

duration_in min/ms	0	0.0
time_signature	0	0.0
Class	0	0.0

In [11]:
```
nullPercentages=nullPercentages.sort_values(['Number of Null Values'],ascending=False)
nullPercentages.head(3)
```

Out[11]:

	Number of Null Values	Fraction of Null Percentages
instrumentalness	4377	24.0
key	2014	11.0
Popularity	428	2.0

In [12]:
```
df[df['Popularity'].isnull()]
```

Out[12]:

	Artist Name	Track Name	Popularity	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	I
25	IndianRaga, Akshay Anantapadmanabhan, Madhu Iy...	Swagatham Krishna - Mohanam - Adi	NaN	0.548	0.711	10.0	-8.440	1	0.1070	0.542000	0.000096	
43	Lilly Wood and The Prick	A Song	NaN	0.618	0.438	11.0	-7.495	1	0.0762	0.407000	NaN	
51	How Lucky	Kurt Vile, John Prine	NaN	0.523	0.441	5.0	-9.928	1	0.0311	0.465000	0.002680	
95	The Beatles	She's Leaving Home - Remastered	NaN	0.381	0.214	4.0	-11.629	1	0.0337	0.944000	NaN	
111	Faith No More	Falling to Pieces	NaN	0.545	0.919	2.0	-7.679	1	0.0483	0.002690	0.005120	
...	...	...	...	...	...	...	...	...	...	...	...	
17741	Westlife	That's Where You Find Love	NaN	0.464	0.493	2.0	-5.626	1	0.0302	0.273000	NaN	
17855	Avarus	Sadist	NaN	0.502	0.991	6.0	-4.612	1	0.1350	0.000173	0.000017	
17873	Erika de Casier	Someone to Chill With	NaN	0.610	0.757	2.0	-6.271	0	0.0377	0.075400	0.010600	
17912	Hype Lights	Something Wrong	NaN	0.385	0.983	4.0	-4.052	0	0.1060	0.000098	0.000010	
17931	Westlife	Drive (For All Time)	NaN	0.585	0.457	9.0	-6.777	1	0.0329	0.158000	NaN	

428 rows × 17 columns

In [13]:
```
nullPopularity=df[df['Popularity'].isnull()]
nullPopularity.index
```

Out[13]:
```
Int64Index([   25,    43,    51,    95,   111,   116,   268,   284,   318,
              335,
            ...
            17626, 17635, 17689, 17712, 17726, 17741, 17855, 17873, 17912,
            17931],
           dtype='int64', length=428)
```

In [14]:
```
df=df.drop(nullPopularity.index)
df
```

Out[14]:

	Artist Name	Track Name	Popularity	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness
0	Bruno Mars	That's What I Like (feat. Gucci Mane)	60.0	0.854	0.564	1.0	-4.964	1	0.0485	0.017100	NaN	0.0849
1	Boston	Hitch a Ride	54.0	0.382	0.814	3.0	-7.230	1	0.0406	0.001100	0.004010	0.1010

2	The Raincoats	No Side to Fall In	35.0	0.434	0.614	6.0	-8.334	1	0.0525	0.486000	0.000196	0.3940
3	Deno	Lingo (feat. J.I & Chunkz)	66.0	0.853	0.597	10.0	-6.528	0	0.0555	0.021200	NaN	0.1220
4	Red Hot Chili Peppers	Nobody Weird Like Me - Remastered	53.0	0.167	0.975	2.0	-4.279	1	0.2160	0.000169	0.016100	0.1720
...	...	...	...	...	...	...	...	...	...	...	...	...
17991	Green-House	Find Home	35.0	0.166	0.109	7.0	-17.100	0	0.0413	0.993000	0.824000	0.0984
17992	Micatone	All Gone	27.0	0.638	0.223	11.0	-10.174	0	0.0329	0.858000	0.000016	0.0705
17993	Smash Hit Combo	Peine perdue	34.0	0.558	0.981	4.0	-4.683	0	0.0712	0.000030	0.000136	0.6660
17994	Beherit	Salomon's Gate	29.0	0.215	0.805	6.0	-12.757	0	0.1340	0.001290	0.916000	0.2560
17995	The Raconteurs	Broken Boy Soldier	43.0	0.400	0.853	4.0	-5.320	0	0.0591	0.006040	0.212000	0.3340

17568 rows × 17 columns

In [15]:
```
nullkey=df[df['key'].isnull()]
nullkey.index
df=df.drop(nullkey.index)
df
```

Out[15]:

	Artist Name	Track Name	Popularity	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness
0	Bruno Mars	That's What I Like (feat. Gucci Mane)	60.0	0.854	0.564	1.0	-4.964	1	0.0485	0.017100	NaN	0.0849
1	Boston	Hitch a Ride	54.0	0.382	0.814	3.0	-7.230	1	0.0406	0.001100	0.004010	0.1010
2	The Raincoats	No Side to Fall In	35.0	0.434	0.614	6.0	-8.334	1	0.0525	0.486000	0.000196	0.3940
3	Deno	Lingo (feat. J.I & Chunkz)	66.0	0.853	0.597	10.0	-6.528	0	0.0555	0.021200	NaN	0.1220
4	Red Hot Chili Peppers	Nobody Weird Like Me - Remastered	53.0	0.167	0.975	2.0	-4.279	1	0.2160	0.000169	0.016100	0.1720
...	...	...	...	...	...	...	...	...	...	...	...	...
17991	Green-House	Find Home	35.0	0.166	0.109	7.0	-17.100	0	0.0413	0.993000	0.824000	0.0984
17992	Micatone	All Gone	27.0	0.638	0.223	11.0	-10.174	0	0.0329	0.858000	0.000016	0.0705
17993	Smash Hit Combo	Peine perdue	34.0	0.558	0.981	4.0	-4.683	0	0.0712	0.000030	0.000136	0.6660
17994	Beherit	Salomon's Gate	29.0	0.215	0.805	6.0	-12.757	0	0.1340	0.001290	0.916000	0.2560
17995	The Raconteurs	Broken Boy Soldier	43.0	0.400	0.853	4.0	-5.320	0	0.0591	0.006040	0.212000	0.3340

15613 rows × 17 columns

In [16]:
```
nullPercentages=pd.DataFrame(df.isnull().sum(),columns=['Number of Null Values'])
nullPercentages['Fraction of Null Percentages']=np.round(df.isnull().sum()/df.shape[0],2)*100
nullPercentages
nullPercentages=nullPercentages.sort_values(['Number of Null Values'],ascending=False)
nullPercentages.head(3)
```

Out[16]:

	Number of Null Values	Fraction of Null Percentages
instrumentalness	3800	24.0
Artist Name	0	0.0
acousticness	0	0.0

In [17]:
```
avg=np.mean(df['instrumentalness'])
avg
```

Out[17]:
```
0.17899141338186741
```

In [18]:
```
df.fillna(value=avg,inplace=True)
df
```

Out[18]:

	Artist Name	Track Name	Popularity	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness
0	Bruno Mars	That's What I Like (feat. Gucci Mane)	60.0	0.854	0.564	1.0	-4.964	1	0.0485	0.017100	0.178991	0.0849
1	Boston	Hitch a Ride	54.0	0.382	0.814	3.0	-7.230	1	0.0406	0.001100	0.004010	0.1010
2	The Raincoats	No Side to Fall In	35.0	0.434	0.614	6.0	-8.334	1	0.0525	0.486000	0.000196	0.3940
3	Deno	Lingo (feat. J.I & Chunkz)	66.0	0.853	0.597	10.0	-6.528	0	0.0555	0.021200	0.178991	0.1220
4	Red Hot Chili Peppers	Nobody Weird Like Me - Remastered	53.0	0.167	0.975	2.0	-4.279	1	0.2160	0.000169	0.016100	0.1720
...	...	...	...	...	...	...	...	...	...	...	...	...
17991	Green-House	Find Home	35.0	0.166	0.109	7.0	-17.100	0	0.0413	0.993000	0.824000	0.0984
17992	Micatone	All Gone	27.0	0.638	0.223	11.0	-10.174	0	0.0329	0.858000	0.000016	0.0705
17993	Smash Hit Combo	Peine perdue	34.0	0.558	0.981	4.0	-4.683	0	0.0712	0.000030	0.000136	0.6660
17994	Beherit	Salomon's Gate	29.0	0.215	0.805	6.0	-12.757	0	0.1340	0.001290	0.916000	0.2560
17995	The Raconteurs	Broken Boy Soldier	43.0	0.400	0.853	4.0	-5.320	0	0.0591	0.006040	0.212000	0.3340

15613 rows × 17 columns

In [19]:
```
nullPercentages=pd.DataFrame(df.isnull().sum(),columns=['Number of Null Values'])
nullPercentages['Fraction of Null Percentages']=np.round(df.isnull().sum()/df.shape[0],2)*100
nullPercentages
```

Out[19]:

	Number of Null Values	Fraction of Null Percentages
Artist Name	0	0.0
Track Name	0	0.0
Popularity	0	0.0
danceability	0	0.0
energy	0	0.0
key	0	0.0
loudness	0	0.0
mode	0	0.0
speechiness	0	0.0
acousticness	0	0.0
instrumentalness	0	0.0
liveness	0	0.0
valence	0	0.0
tempo	0	0.0
duration_in min/ms	0	0.0
time_signature	0	0.0
Class	0	0.0

In [20]:
```
artist=df['Artist Name'].unique()
list(artist)
```

```
len(artist)
```

Out[20]: 8264

In [21]:
```
collabrationLambda=lambda x:1 if(len(x.split(',')))>1 else 0
```

In [22]:
```
df['Collabration']=df['Artist Name'].apply(collabrationLambda)
df
```

Out[22]:

	Artist Name	Track Name	Popularity	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness
0	Bruno Mars	That's What I Like (feat. Gucci Mane)	60.0	0.854	0.564	1.0	-4.964	1	0.0485	0.017100	0.178991	0.0849
1	Boston	Hitch a Ride	54.0	0.382	0.814	3.0	-7.230	1	0.0406	0.001100	0.004010	0.1010
2	The Raincoats	No Side to Fall In	35.0	0.434	0.614	6.0	-8.334	1	0.0525	0.486000	0.000196	0.3940
3	Deno	Lingo (feat. J.I & Chunkz)	66.0	0.853	0.597	10.0	-6.528	0	0.0555	0.021200	0.178991	0.1220
4	Red Hot Chili Peppers	Nobody Weird Like Me - Remastered	53.0	0.167	0.975	2.0	-4.279	1	0.2160	0.000169	0.016100	0.1720
...	...	...	...	...	...	...	...	...	...	...	...	...
17991	Green-House	Find Home	35.0	0.166	0.109	7.0	-17.100	0	0.0413	0.993000	0.824000	0.0984
17992	Micatone	All Gone	27.0	0.638	0.223	11.0	-10.174	0	0.0329	0.858000	0.000016	0.0705
17993	Smash Hit Combo	Peine perdue	34.0	0.558	0.981	4.0	-4.683	0	0.0712	0.000030	0.000136	0.6660
17994	Beherit	Salomon's Gate	29.0	0.215	0.805	6.0	-12.757	0	0.1340	0.001290	0.916000	0.2560
17995	The Raconteurs	Broken Boy Soldier	43.0	0.400	0.853	4.0	-5.320	0	0.0591	0.006040	0.212000	0.3340

15613 rows × 18 columns

ExploratoryDataAnalaaysis(EDA)

In [23]:
```
sns.set()
```

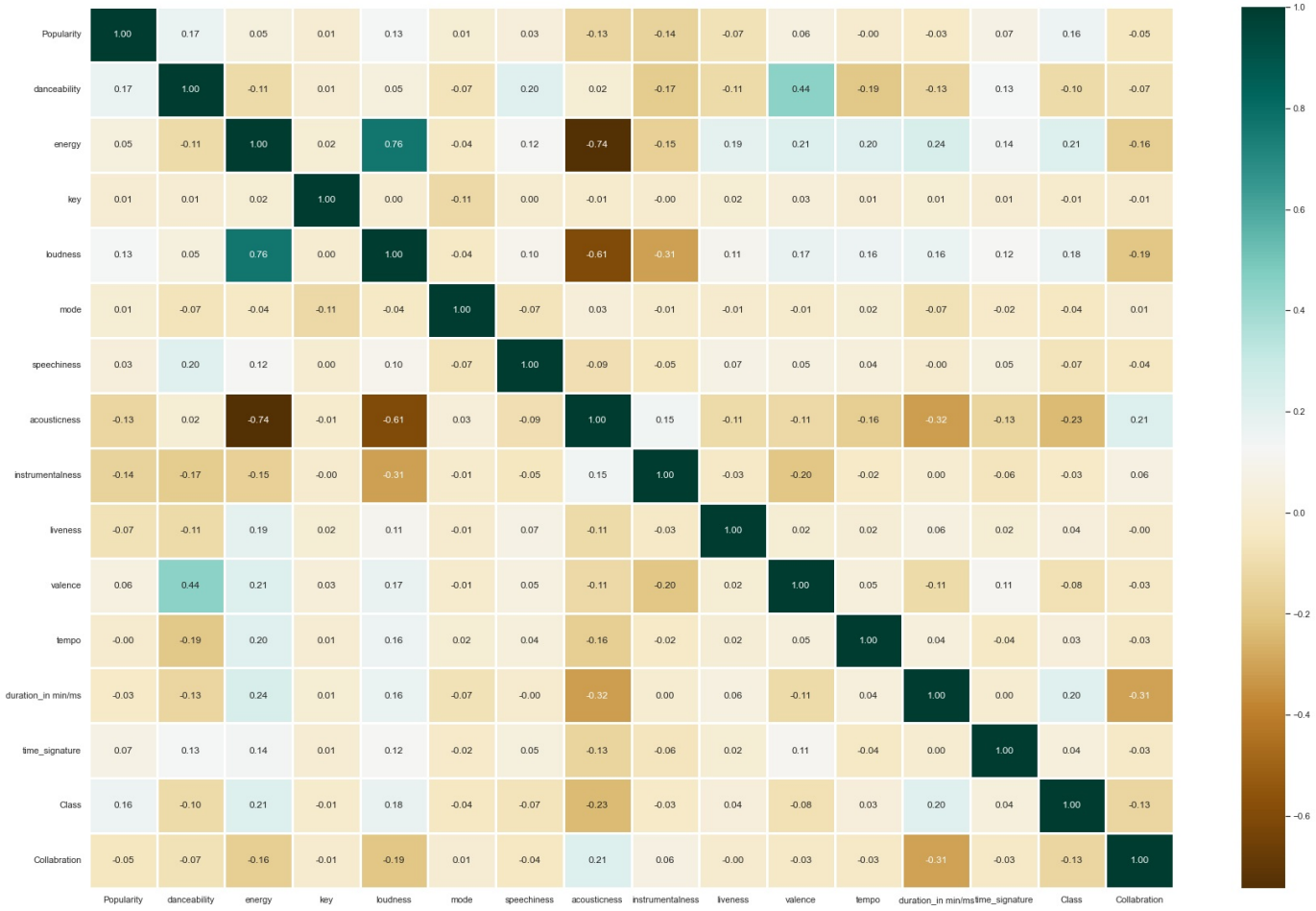
In [24]:
```
df.corr()
```

Out[24]:

	Popularity	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	
Popularity	1.000000	0.167841	0.049912	0.005901	0.128904	0.014927	0.034451	-0.128204	-0.141583	-0.074121	0
danceability	0.167841	1.000000	-0.107789	0.007082	0.049396	-0.068680	0.203774	0.017916	-0.172828	-0.105875	0
energy	0.049912	-0.107789	1.000000	0.015435	0.762748	-0.036950	0.124470	-0.743266	-0.151946	0.191197	0
key	0.005901	0.007082	0.015435	1.000000	0.002787	-0.111741	0.001881	-0.009717	-0.001664	0.015091	0
loudness	0.128904	0.049396	0.762748	0.002787	1.000000	-0.038943	0.095718	-0.605532	-0.307774	0.105653	0
mode	0.014927	-0.068680	-0.036950	-0.111741	-0.038943	1.000000	-0.067925	0.026868	-0.012555	-0.006292	-0
speechiness	0.034451	0.203774	0.124470	0.001881	0.095718	-0.067925	1.000000	-0.086722	-0.052040	0.071345	0
acousticness	-0.128204	0.017916	-0.743266	-0.009717	-0.605532	0.026868	-0.086722	1.000000	0.147989	-0.107990	-0
instrumentalness	-0.141583	-0.172828	-0.151946	-0.001664	-0.307774	-0.012555	-0.052040	0.147989	1.000000	-0.034856	-0
liveness	-0.074121	-0.105875	0.191197	0.015091	0.105653	-0.006292	0.071345	-0.107990	-0.034856	1.000000	0
valence	0.055601	0.444102	0.210236	0.030430	0.170883	-0.009946	0.054440	-0.114044	-0.197926	0.015657	1
tempo	-0.004236	-0.187918	0.204699	0.012491	0.157005	0.016785	0.043878	-0.161297	-0.021640	0.023245	0
duration_in min/ms	-0.028111	-0.128275	0.241118	0.013494	0.163787	-0.069994	-0.002812	-0.320331	0.004829	0.055559	-0
time_signature	0.067760	0.129610	0.138418	0.007617	0.122890	-0.016428	0.053313	-0.131312	-0.061119	0.016813	0

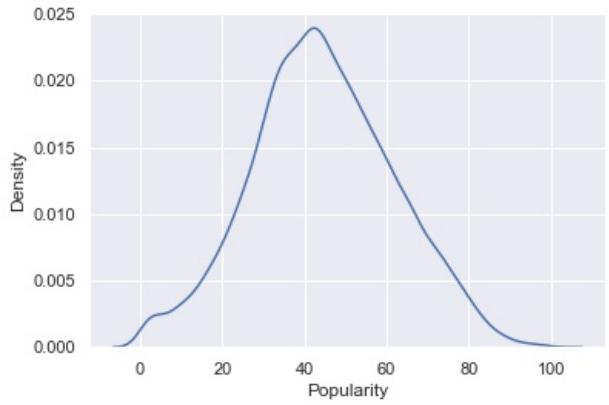
Class	0.156755	-0.104809	0.210718	-0.008763	0.175312	-0.044157	-0.067720	-0.233007	-0.026554	0.038376	-0
Collabration	-0.051961	-0.070131	-0.158281	-0.006641	-0.194197	0.009623	-0.043128	0.213029	0.064233	-0.004538	-0

In [25]:
```
plt.figure(figsize=(30,20))
sns.heatmap(df.corr(),fmt='.2f',cmap='BrBG',
            linewidth=3,annot=True,xticklabels=True)
plt.show()
```



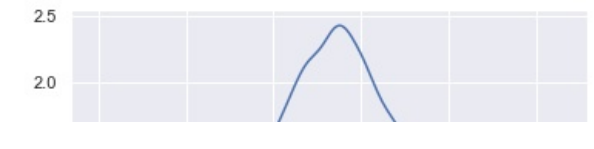
In [26]:
```
sns.kdeplot(df['Popularity'])
```

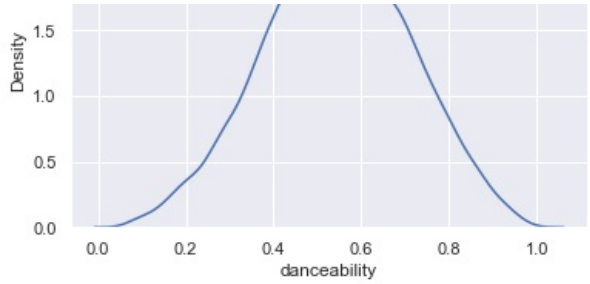
Out[26]:
```
<AxesSubplot:xlabel='Popularity', ylabel='Density'>
```



In [27]:
```
sns.kdeplot(df['danceability'])
```

Out[27]:
```
<AxesSubplot:xlabel='danceability', ylabel='Density'>
```



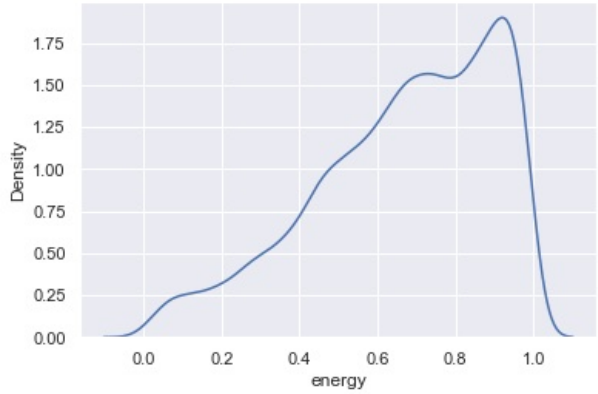


In [28]:
```
df.columns
```

Out[28]:
```
Index(['Artist Name', 'Track Name', 'Popularity', 'danceability', 'energy',
       'key', 'loudness', 'mode', 'speechiness', 'acousticness',
       'instrumentalness', 'liveness', 'valence', 'tempo',
       'duration_in min/ms', 'time_signature', 'Class', 'Collabration'],
      dtype='object')
```

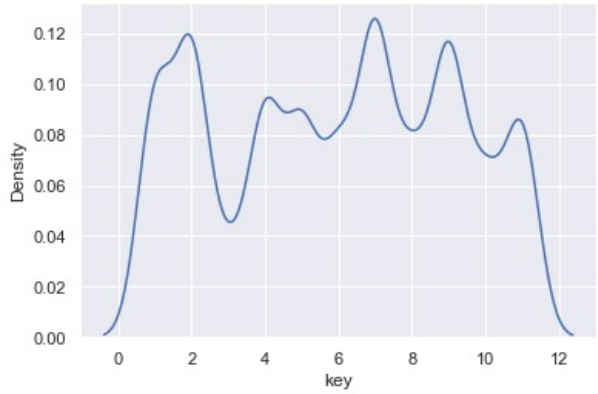
In [29]:
```
sns.kdeplot(df['energy'])
```

Out[29]:
```
<AxesSubplot:xlabel='energy', ylabel='Density'>
```



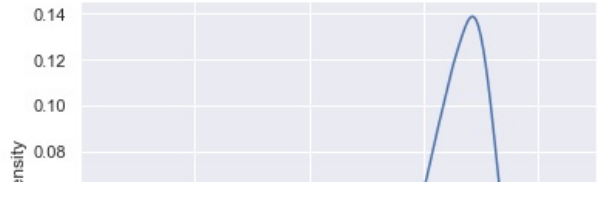
In [30]:
```
sns.kdeplot(df['key'])
```

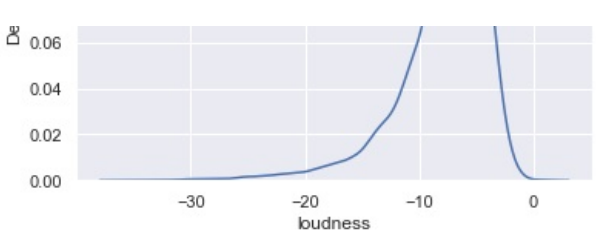
Out[30]:
```
<AxesSubplot:xlabel='key', ylabel='Density'>
```



In [31]:
```
sns.kdeplot(df['loudness'])
```

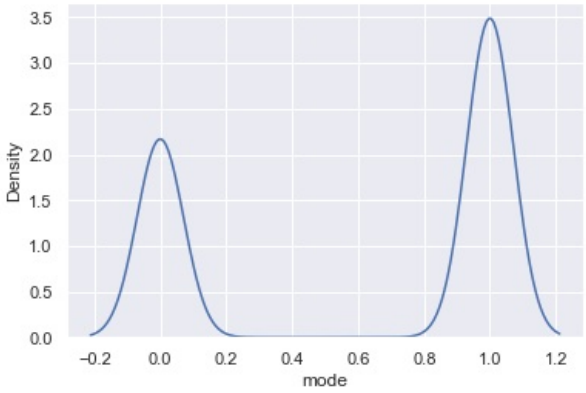
Out[31]:
```
<AxesSubplot:xlabel='loudness', ylabel='Density'>
```





In [32]:
```
sns.kdeplot(df['mode'])
```

Out[32]:
```
<AxesSubplot:xlabel='mode', ylabel='Density'>
```

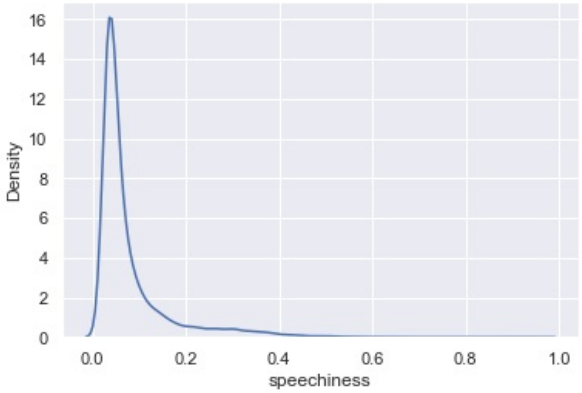


In [33]:
```
df['mode'].unique()
```

Out[33]:
```
array([1, 0], dtype=int64)
```

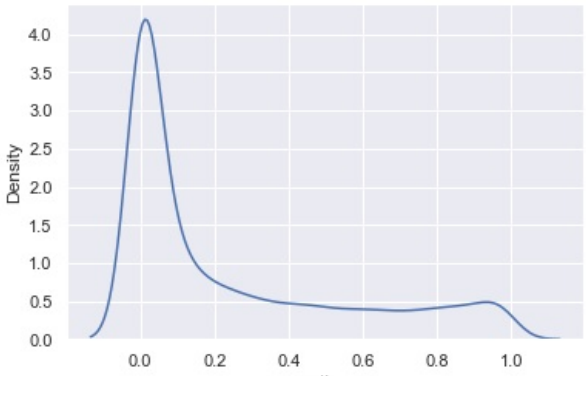
In [34]:
```
sns.kdeplot(df['speechiness'])
```

Out[34]:
```
<AxesSubplot:xlabel='speechiness', ylabel='Density'>
```



In [35]:
```
sns.kdeplot(df['acousticness'])
```

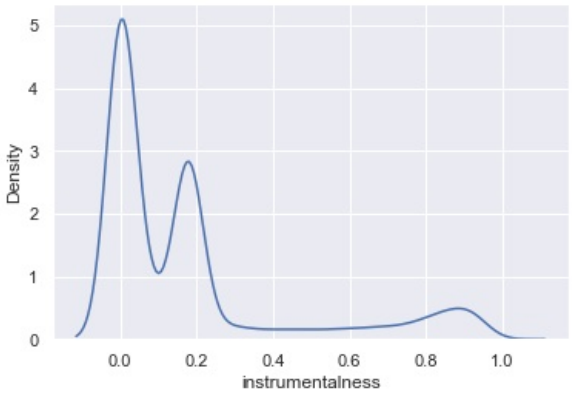
Out[35]:
```
<AxesSubplot:xlabel='acousticness', ylabel='Density'>
```



acousticness

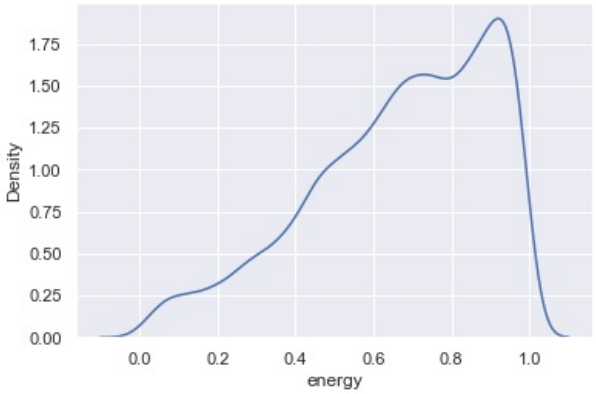
In [36]:
```
sns.kdeplot(df['instrumentalness'])
```

Out[36]:
```
<AxesSubplot:xlabel='instrumentalness', ylabel='Density'>
```



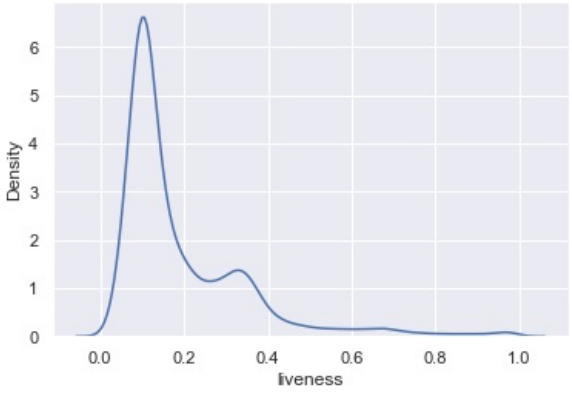
In [37]:
```
sns.kdeplot(df['energy'])
```

Out[37]:
```
<AxesSubplot:xlabel='energy', ylabel='Density'>
```



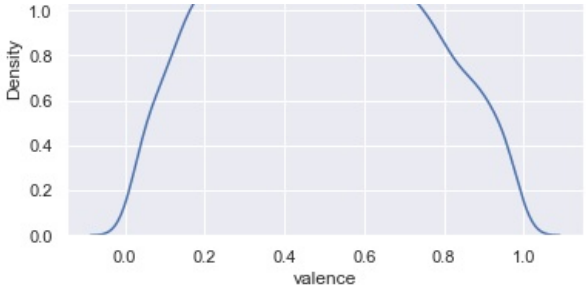
In [38]:
```
sns.kdeplot(df['liveness'])
```

Out[38]:
```
<AxesSubplot:xlabel='liveness', ylabel='Density'>
```



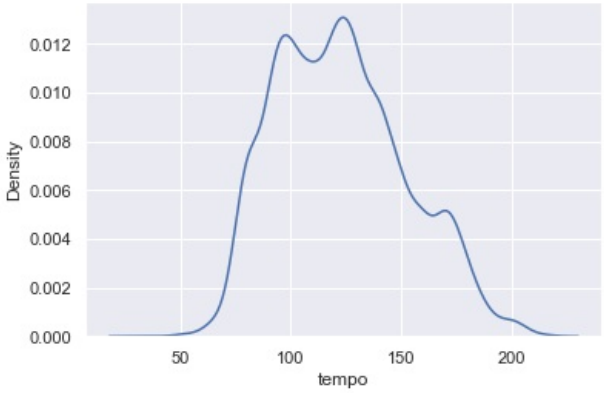
In [39]:
```
sns.kdeplot(df['valence'])
```

Out[39]:
```
<AxesSubplot:xlabel='valence', ylabel='Density'>
```



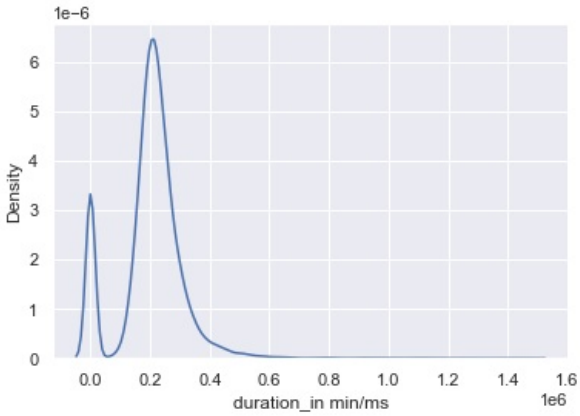
In [40]:
```
sns.kdeplot(df['tempo'])
```

Out[40]:
```
<AxesSubplot:xlabel='tempo', ylabel='Density'>
```



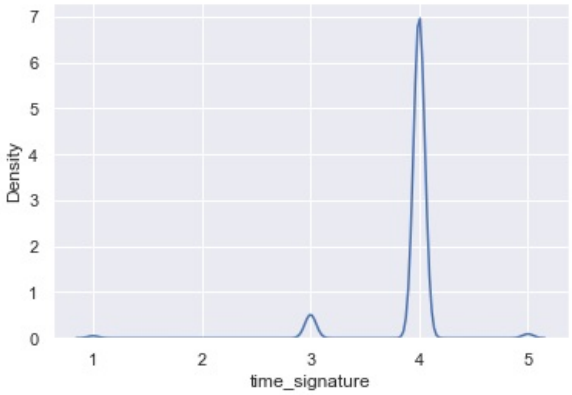
In [41]:
```
sns.kdeplot(df[ 'duration_in min/ms'])
```

Out[41]:
```
<AxesSubplot:xlabel='duration_in min/ms', ylabel='Density'>
```



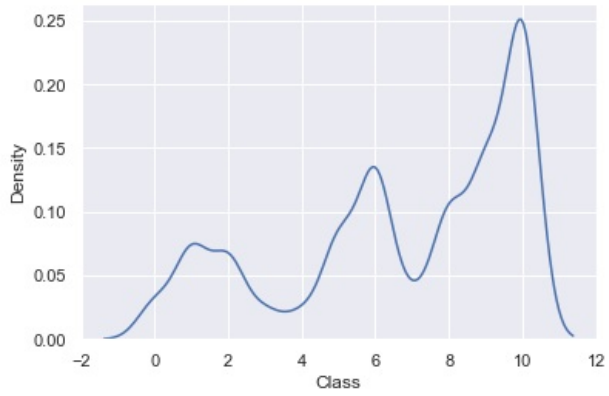
In [42]:
```
sns.kdeplot(df['time_signature'])
```

Out[42]:
```
<AxesSubplot:xlabel='time_signature', ylabel='Density'>
```



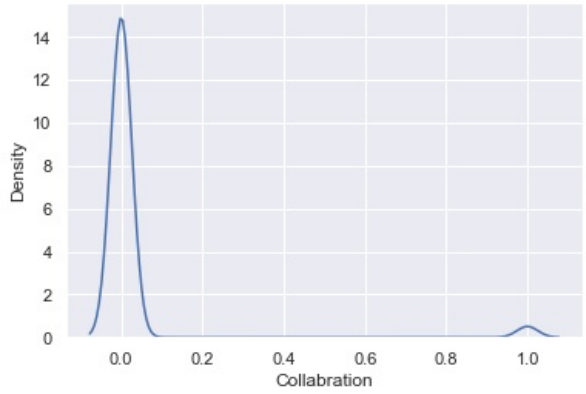
In [43]:
```
sns.kdeplot(df['Class'])
```

Out[43]:
```
<AxesSubplot:xlabel='Class', ylabel='Density'>
```



In [44]:
```
sns.kdeplot(df['Collabration'])
```

Out[44]:
```
<AxesSubplot:xlabel='Collabration', ylabel='Density'>
```



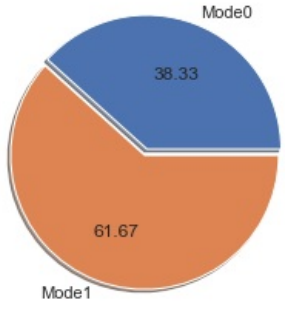
In [45]:
```
count1=df[df['mode']==1]
count1=count1.shape[0]
count1
```

Out[45]:
```
9628
```

In [46]:
```
count0=df[df['mode']==0]
count0=count0.shape[0]
count0
```

Out[46]:
```
5985
```

In [47]:
```
plt.pie([count0,count1],labels=['Mode0','Mode1'],autopct='%.2f',shadow=True,explode=[0.05,0])
plt.show()
```



In [48]:
```
count0=df[df['Class']==0].shape[0]
count0
```

Out[48]:
```
491
```

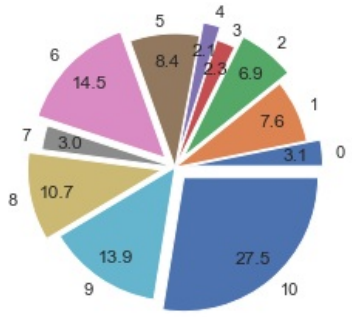
In [49]:
```
a=df['Class'].unique()
len(a)
a
```

Out[49]:
```
array([ 5, 10,  6,  2,  4,  8,  9,  3,  7,  1,  0], dtype=int64)
```

In [50]:
```
count1=df[df['Class']==1].shape[0]
count1
count2=df[df['Class']==2].shape[0]
count2
count3=df[df['Class']==3].shape[0]
count3
count4=df[df['Class']==4].shape[0]
count4
count5=df[df['Class']==5].shape[0]
count5
count6=df[df['Class']==6].shape[0]
count6
count7=df[df['Class']==7].shape[0]
count7
count8=df[df['Class']==8].shape[0]
count8
count9=df[df['Class']==9].shape[0]
count9
count10=df[df['Class']==10].shape[0]
count10
```

Out[50]:
```
4298
```

In [51]:
```
plt.pie([count0,count1,count2,count3,count4,count5,count6,count7,count8,count9,count10],labels=['0','1','2','3','
plt.show()
```



In [52]:
```
count=[]
for i in range(len(df['Class'].unique())):
    count.append(df[df['Class']==i].shape[0])
count
```

Out[52]:
```
[491, 1192, 1075, 358, 326, 1311, 2263, 465, 1664, 2170, 4298]
```

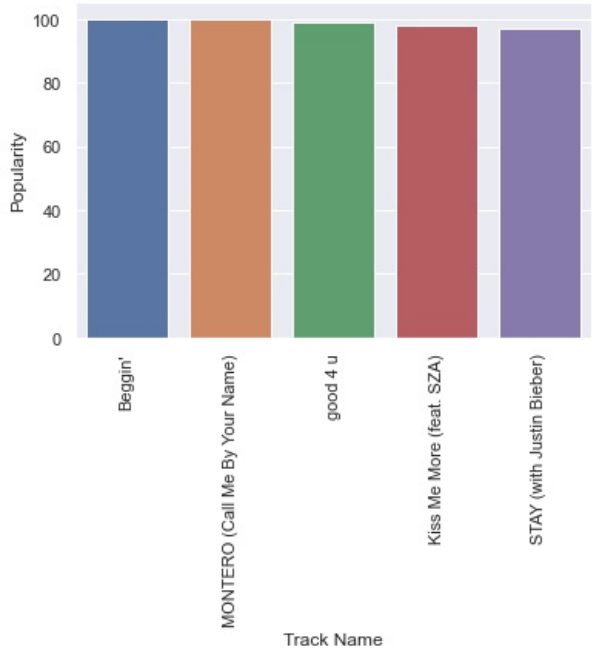
In [53]:
```
top=df.sort_values(by=['Popularity'],ascending=False)
top5=top.head(5)
top5
```

Out[53]:

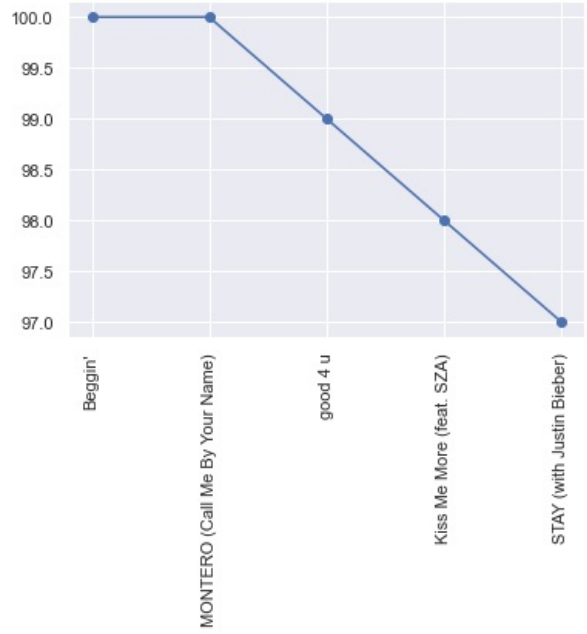
	Artist Name	Track Name	Popularity	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	va
13489	Måneskin	Beggin'	100.0	0.714	0.800	11.0	-4.808	0	0.0504	0.1270	0.178991	0.3590	
		MONTERO											

11781	Lil Nas X	(Call Me By Your Name)	100.0	0.610	0.508	8.0	-6.682	0	0.1520	0.2970	0.178991	0.3840
187	Olivia Rodrigo	good 4 u	99.0	0.563	0.664	9.0	-5.044	1	0.1540	0.3350	0.178991	0.0849
953	Doja Cat	Kiss Me More (feat. SZA)	98.0	0.762	0.701	8.0	-3.541	1	0.0286	0.2350	0.000158	0.1230
14830	The Kid LAROI, Justin Bieber	STAY (with Justin Bieber)	97.0	0.591	0.764	1.0	-5.484	1	0.0483	0.0383	0.178991	0.1030

In [54]:
```
sns.barplot(top5['Track Name'],top5['Popularity'])
plt.xticks(rotation=90)
plt.show()
```



In [55]:
```
plt.plot(top5['Track Name'],top5['Popularity'])
plt.scatter(top5['Track Name'],top5['Popularity'])
plt.xticks(rotation=90)
plt.show()
```



In [56]:
```
long=df.sort_values(by=['duration_in min/ms'],ascending=False)
long5=long.head(5)
```

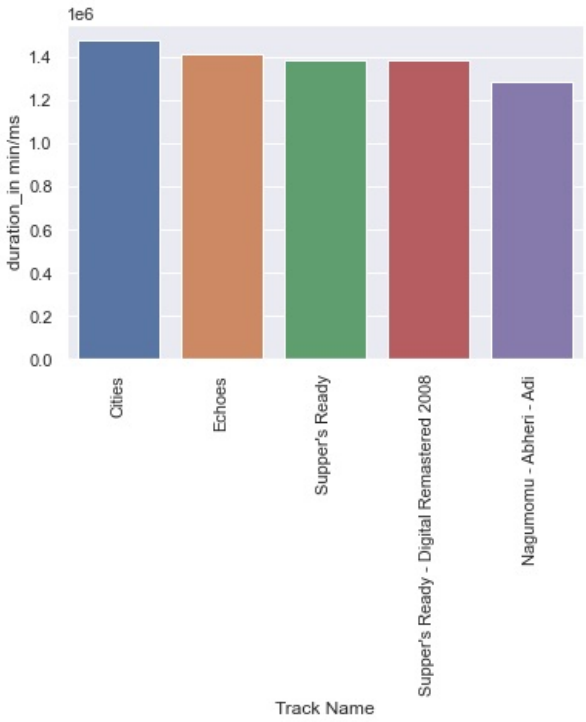
```
long5
```

Out[56]:

	Artist Name	Track Name	Popularity	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	vale
14837	Phish	Cities	17.0	0.432	0.651	7.0	-7.443	1	0.0325	0.319	0.164000	0.7010	0
2174	Pink Floyd	Echoes	57.0	0.285	0.322	6.0	-16.580	0	0.0440	0.366	0.651000	0.1070	0
10054	Genesis	Supper's Ready	43.0	0.310	0.610	4.0	-9.415	0	0.0723	0.147	0.000476	0.1790	0
13222	Genesis	Supper's Ready - Digital Remastered 2008	43.0	0.310	0.610	4.0	-9.415	0	0.0723	0.147	0.000476	0.1790	0
2028	Rajhesh Vaidhya	Nagumomu - Abheri - Adi	19.0	0.395	0.774	2.0	-7.731	0	0.0433	0.346	0.841000	0.0898	0

In [57]:

```
sns.barplot(long5['Track Name'],long5['duration_in min/ms'])
plt.xticks(rotation=90)
plt.show()
```



In [58]:

```
df['energy'].unique()
```

Out[58]:

```
array([0.564 , 0.814 , 0.614 , ..., 0.0773, 0.0995, 0.0835])
```

In [59]:

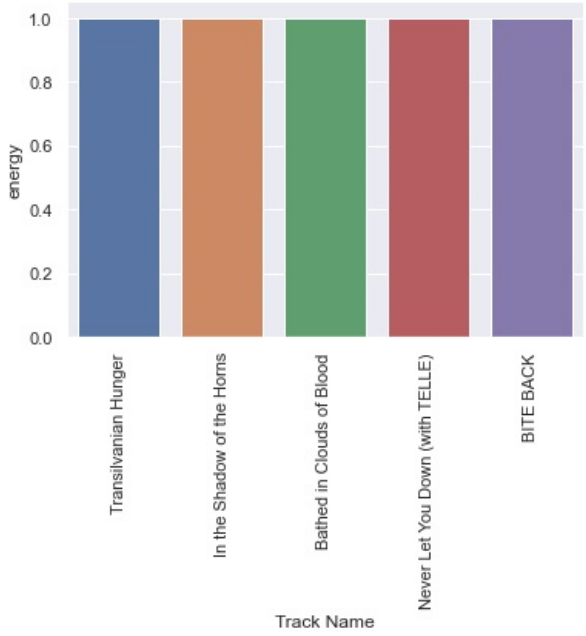
```
eng=df.sort_values(by=['energy'],ascending=False)
eng5=eng.head(5)
eng5
```

Out[59]:

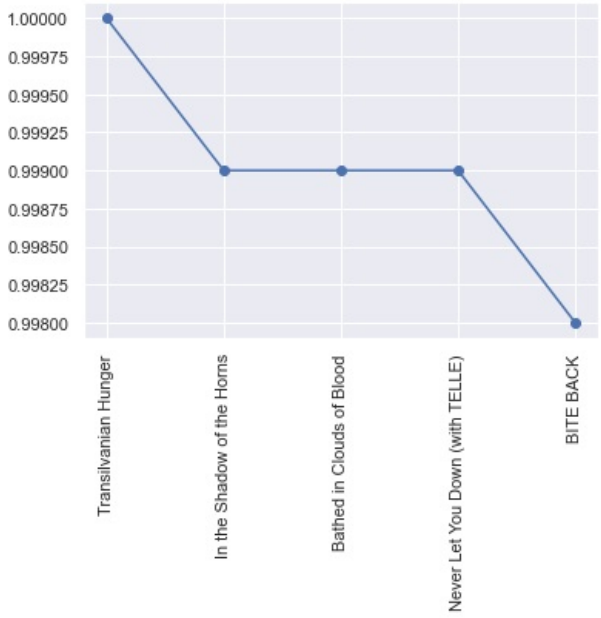
	Artist Name	Track Name	Popularity	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness
15433	Darkthrone	Transilvanian Hunger	47.0	0.267	1.000	4.0	-4.920	0	0.1350	0.016000	0.447000	0.450
5914	Darkthrone	In the Shadow of the Horns	41.0	0.141	0.999	11.0	-5.335	1	0.1400	0.000004	0.008040	0.109
17274	Judas Iscariot	Bathed in Clouds of Blood	30.0	0.189	0.999	7.0	-9.303	0	0.0656	0.000041	0.785000	0.304
13063	Kayzo	Never Let You Down (with TELLE)	44.0	0.446	0.999	9.0	-1.590	0	0.3130	0.028900	0.056600	0.612

3653	FEVER 333	BITE BACK	50.0	0.545	0.998	8.0	-3.645	1	0.2340	0.019200	0.178991	0.877

In [60]:
```
sns.barplot(eng5['Track Name'],eng5['energy'])
plt.xticks(rotation=90)
plt.show()
```



In [61]:
```
plt.plot(eng5['Track Name'],eng5['energy'])
plt.scatter(eng5['Track Name'],eng5['energy'])
plt.xticks(rotation=90)
plt.show()
```



MODEL TRAINING AND BUILDING

In [62]:
```
df.reset_index(drop=True,inplace=True)
df
```

Out[62]:

	Artist Name	Track Name	Popularity	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness
0	Bruno Mars	That's What I Like (feat. Gucci Mane)	60.0	0.854	0.564	1.0	-4.964	1	0.0485	0.017100	0.178991	0.0849

1	Boston	Hitch a Ride	54.0	0.382	0.814	3.0	-7.230	1	0.0406	0.001100	0.004010	0.1010
2	The Raincoats	No Side to Fall In	35.0	0.434	0.614	6.0	-8.334	1	0.0525	0.486000	0.000196	0.3940
3	Deno	Lingo (feat. J.I & Chunkz)	66.0	0.853	0.597	10.0	-6.528	0	0.0555	0.021200	0.178991	0.1220
4	Red Hot Chili Peppers	Nobody Weird Like Me - Remastered	53.0	0.167	0.975	2.0	-4.279	1	0.2160	0.000169	0.016100	0.1720
...	...	...	...	...	...	...	...	...	...	...	...	...
15608	Green-House	Find Home	35.0	0.166	0.109	7.0	-17.100	0	0.0413	0.993000	0.824000	0.0984
15609	Micatone	All Gone	27.0	0.638	0.223	11.0	-10.174	0	0.0329	0.858000	0.000016	0.0705
15610	Smash Hit Combo	Peine perdue	34.0	0.558	0.981	4.0	-4.683	0	0.0712	0.000030	0.000136	0.6660
15611	Beherit	Salomon's Gate	29.0	0.215	0.805	6.0	-12.757	0	0.1340	0.001290	0.916000	0.2560
15612	The Raconteurs	Broken Boy Soldier	43.0	0.400	0.853	4.0	-5.320	0	0.0591	0.006040	0.212000	0.3340

15613 rows × 18 columns

In [63]:
```
df.columns
```

Out[63]:
```
Index(['Artist Name', 'Track Name', 'Popularity', 'danceability', 'energy',
       'key', 'loudness', 'mode', 'speechiness', 'acousticness',
       'instrumentalness', 'liveness', 'valence', 'tempo',
       'duration_in min/ms', 'time_signature', 'Class', 'Collabration'],
      dtype='object')
```

In [64]:
```
df2=df.drop(['Artist Name','Track Name','mode','time_signature','Class','Collabration'],axis=1)
df2
```

Out[64]:

	Popularity	danceability	energy	key	loudness	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_in min/ms
0	60.0	0.854	0.564	1.0	-4.964	0.0485	0.017100	0.178991	0.0849	0.8990	134.071	234596.0
1	54.0	0.382	0.814	3.0	-7.230	0.0406	0.001100	0.004010	0.1010	0.5690	116.454	251733.0
2	35.0	0.434	0.614	6.0	-8.334	0.0525	0.486000	0.000196	0.3940	0.7870	147.681	109667.0
3	66.0	0.853	0.597	10.0	-6.528	0.0555	0.021200	0.178991	0.1220	0.5690	107.033	173968.0
4	53.0	0.167	0.975	2.0	-4.279	0.2160	0.000169	0.016100	0.1720	0.0918	199.060	229960.0
...	...	...	...	...	...	...	...	...	...	...	...	...
15608	35.0	0.166	0.109	7.0	-17.100	0.0413	0.993000	0.824000	0.0984	0.1770	171.587	193450.0
15609	27.0	0.638	0.223	11.0	-10.174	0.0329	0.858000	0.000016	0.0705	0.3350	73.016	257067.0
15610	34.0	0.558	0.981	4.0	-4.683	0.0712	0.000030	0.000136	0.6660	0.2620	105.000	216222.0
15611	29.0	0.215	0.805	6.0	-12.757	0.1340	0.001290	0.916000	0.2560	0.3550	131.363	219693.0
15612	43.0	0.400	0.853	4.0	-5.320	0.0591	0.006040	0.212000	0.3340	0.3770	138.102	182227.0

15613 rows × 12 columns

In [65]:
```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
```

In [66]:
```
df2=scaler.fit_transform(df2)
```

In [67]:
```
df2=pd.DataFrame(df2,columns=[ 'Popularity', 'danceability', 'energy',
       'key', 'loudness',  'speechiness', 'acousticness',
       'instrumentalness', 'liveness', 'valence', 'tempo',
       'duration_in min/ms'])
df2
```

Out[67]:

	Popularity	danceability	energy	key	loudness	speechiness	acousticness	instrumentalness	liveness	valence	tempo	d

0	0.892192	1.862378	-0.441135	-1.551717	0.727199	-0.385493	-0.731701	-1.050342e-16	-0.701987	1.725650	0.387108
1	0.546338	-0.971474	0.629952	-0.925698	0.158830	-0.478702	-0.783482	-6.621742e-01	-0.599960	0.349115	-0.209585
2	-0.548867	-0.659270	-0.226918	0.013332	-0.118080	-0.338298	0.785790	-6.766073e-01	1.256803	1.258462	0.848083
3	1.238046	1.856374	-0.299751	1.265371	0.334909	-0.302902	-0.718433	-1.050342e-16	-0.466882	0.349115	-0.528676
4	0.488695	-2.262318	1.319732	-1.238708	0.899014	1.590773	-0.786495	-6.164225e-01	-0.150028	-1.641438	2.588303
...	...	...	...	...	...	...	...	...	...	...	...
15608	-0.548867	-2.268322	-2.390514	0.326342	-2.316809	-0.470443	2.426584	2.440876e+00	-0.616437	-1.286041	1.657785
15609	-1.010005	0.565530	-1.902098	1.578381	-0.579597	-0.569551	1.989686	-6.772881e-01	-0.793241	-0.626973	-1.680842
15610	-0.606509	0.085216	1.345439	-0.612688	0.797680	-0.117664	-0.786946	-6.768344e-01	2.980487	-0.931479	-0.597535
15611	-0.894721	-1.974130	0.591393	0.013332	-1.227477	0.623288	-0.782867	2.789028e+00	0.382286	-0.543547	0.295387
15612	-0.087728	-0.863404	0.797042	-0.612688	0.637905	-0.260427	-0.767495	1.249129e-01	0.876578	-0.451778	0.523639

15613 rows × 12 columns

In [68]:
```
df=df.drop([ 'Popularity', 'danceability', 'energy',
       'key', 'loudness',  'speechiness', 'acousticness',
       'instrumentalness', 'liveness', 'valence', 'tempo',
       'duration_in min/ms'],axis=1)
```

In [69]:
```
df
```

Out[69]:

	Artist Name	Track Name	mode	time_signature	Class	Collabration
0	Bruno Mars	That's What I Like (feat. Gucci Mane)	1	4	5	0
1	Boston	Hitch a Ride	1	4	10	0
2	The Raincoats	No Side to Fall In	1	4	6	0
3	Deno	Lingo (feat. J.I & Chunkz)	0	4	5	0
4	Red Hot Chili Peppers	Nobody Weird Like Me - Remastered	1	4	10	0
...	...	...	...	...	...	...
15608	Green-House	Find Home	0	3	6	0
15609	Micatone	All Gone	0	4	2	0
15610	Smash Hit Combo	Peine perdue	0	4	8	0
15611	Beherit	Salomon's Gate	0	4	8	0
15612	The Raconteurs	Broken Boy Soldier	0	4	10	0

15613 rows × 6 columns

In [70]:
```
df2
```

Out[70]:

	Popularity	danceability	energy	key	loudness	speechiness	acousticness	instrumentalness	liveness	valence	tempo	d
0	0.892192	1.862378	-0.441135	-1.551717	0.727199	-0.385493	-0.731701	-1.050342e-16	-0.701987	1.725650	0.387108	
1	0.546338	-0.971474	0.629952	-0.925698	0.158830	-0.478702	-0.783482	-6.621742e-01	-0.599960	0.349115	-0.209585	
2	-0.548867	-0.659270	-0.226918	0.013332	-0.118080	-0.338298	0.785790	-6.766073e-01	1.256803	1.258462	0.848083	
3	1.238046	1.856374	-0.299751	1.265371	0.334909	-0.302902	-0.718433	-1.050342e-16	-0.466882	0.349115	-0.528676	
4	0.488695	-2.262318	1.319732	-1.238708	0.899014	1.590773	-0.786495	-6.164225e-01	-0.150028	-1.641438	2.588303	
...	...	...	...	...	...	...	...	...	...	...	...	
15608	-0.548867	-2.268322	-2.390514	0.326342	-2.316809	-0.470443	2.426584	2.440876e+00	-0.616437	-1.286041	1.657785	
15609	-1.010005	0.565530	-1.902098	1.578381	-0.579597	-0.569551	1.989686	-6.772881e-01	-0.793241	-0.626973	-1.680842	
15610	-0.606509	0.085216	1.345439	-0.612688	0.797680	-0.117664	-0.786946	-6.768344e-01	2.980487	-0.931479	-0.597535	
15611	-0.894721	-1.974130	0.591393	0.013332	-1.227477	0.623288	-0.782867	2.789028e+00	0.382286	-0.543547	0.295387	
15612	-0.087728	-0.863404	0.797042	-0.612688	0.637905	-0.260427	-0.767495	1.249129e-01	0.876578	-0.451778	0.523639	

15613 rows × 12 columns

In [71]:
```
df=pd.concat([df,df2],axis=1)
```

In [72]:
```
df
```

Out[72]:

	Artist Name	Track Name	mode	time_signature	Class	Collabration	Popularity	danceability	energy	key	loudness	speechiness
0	Bruno Mars	That's What I Like (feat. Gucci Mane)	1	4	5	0	0.892192	1.862378	-0.441135	-1.551717	0.727199	-0.385493
1	Boston	Hitch a Ride	1	4	10	0	0.546338	-0.971474	0.629952	-0.925698	0.158830	-0.478702
2	The Raincoats	No Side to Fall In	1	4	6	0	-0.548867	-0.659270	-0.226918	0.013332	-0.118080	-0.338298
3	Deno	Lingo (feat. J.I & Chunkz)	0	4	5	0	1.238046	1.856374	-0.299751	1.265371	0.334909	-0.302902
4	Red Hot Chili Peppers	Nobody Weird Like Me - Remastered	1	4	10	0	0.488695	-2.262318	1.319732	-1.238708	0.899014	1.590773
...	...	...	...	...	...	...	...	...	...	...	...	...
15608	Green-House	Find Home	0	3	6	0	-0.548867	-2.268322	-2.390514	0.326342	-2.316809	-0.470443
15609	Micatone	All Gone	0	4	2	0	-1.010005	0.565530	-1.902098	1.578381	-0.579597	-0.569551
15610	Smash Hit Combo	Peine perdue	0	4	8	0	-0.606509	0.085216	1.345439	-0.612688	0.797680	-0.117664
15611	Beherit	Salomon's Gate	0	4	8	0	-0.894721	-1.974130	0.591393	0.013332	-1.227477	0.623288
15612	The Raconteurs	Broken Boy Soldier	0	4	10	0	-0.087728	-0.863404	0.797042	-0.612688	0.637905	-0.260427

15613 rows × 18 columns

In [73]:
```
list(df['Artist Name'].unique())
```

Out[73]:
```
['Bruno Mars',
 'Boston',
 'The Raincoats',
 'Deno',
 'Red Hot Chili Peppers',
 'The Stooges',
 'Solomon Burke',
 'Randy Travis',
 'Professional Murder Music',
 'Dudu Aharon',
 'Mohammed Rafi',
 'Arctic Monkeys',
 'Eyal Golan',
 'Harald Lassen, Bram de Looze',
 'Buffalo Springfield',
 'Elmore James',
 'Velvet Two Stripes',
 'Eden Ben Zaken',
 'DaBaby',
 'Mura Masa',
 'Clairmont The Second',
 'Krankhead',
 'The Stone Roses',
 'R3HAB',
 'MC Hammer',
 'duendita',
 'Rainbow',
 'fijitrip',
 'Machine Gun Kelly',
 'ShittyCT',
 'Tetrarch',
 'Brandon Jack & The Artifacts',
 'Ivri Lider',
 'Vieux Farka Tour√©',
 'Kvelertak',
 'Better Than Ezra',
 'Silversun Pickups',
 'Orange Juice',
 'U2',
 'Omer Adam',
 'Victor Manu',
 'The Last Shadow Puppets',
 'Anne-Marie',
 'Veruca Salt',
 'Juice WRLD',
```

'BENEE',
'Berry Sakharof',
'Ganser',
'Dion',
'Soen',
'Vani Jairam',
'Liam Gallagher',
'The Pretty Things',
'Tha Dogg Pound',
'BLAB',
'Trillary Banks',
'El Goodo',
'Why Bonnie',
'Snoop Dogg',
'MOD SUN',
'Turbo Knight',
'The Black Crowes',
'Radiohead',
'Oceans',
'The Stranglers',
'Itay Levi',
'Joe Bonamassa',
'LeRoi Thirteen',
'Us The Duo',
'Keller Williams',
'Primal Scream',
'Lil Son Jackson',
'Drake',
'Abstract Orchestra',
'Luther Allison',
'beabadoobee',
'R.E.M.',
'FCG Heem',
'Tom The Mail Man',
'ANSON',
'Jimmy Eat World',
'dempsey hope',
'Howie Lee',
'Pit Pony',
'Niko B',
'Ruthi Navon',
'Dan + Shay',
'Gossip',
'Gone Is Gone',
'Violent Soho',
'Mr. Oizo',
'Cage The Elephant',
'Carusella',
'FLO',
'Pathiena',
'Pain of Salvation',
'Black Sabbath',
'The Dark Element',
'Noah Kahan',
'The First Edition',
'Alka Yagnik',
'Bryan Adams',
'The Shins',
'benny blanco',
'The 1975',
'Winger',
'Townes Van Zandt',
'Piers Faccini',
'Athletic Progression',
'Thirty Seconds To Mars',
'Pete Rock & C.L. Smooth',
'Peter Case',
'Menahan Street Band',
'Butcher Babies',
'MICROCORPS',
'Sarah Palu',
'Saltatio Mortis',
'JADHU',
'Bring Me The Horizon',
'Frost*',
'Toto Espinoza',
'T-Bone Walker',
'K. J. Yesudas',
'Jimi Hendrix',
'Eskiz',
'Lowest Creature',
'Tomer Yeshayahu',
'Neon Trees',
'JJ Lin',
'Foreign Air',
'S. P. Balasubrahmanyam',
'Scary Kids Scaring Kids',
'Lawrence of Arcadia',
'Nirvana',

'Breezie 311',
'Ghali',
'Gwar',
'Downset',
'Blue Material',
'Sempra',
'Malcolm McLaren',
'Arcade Fire',
'Bad Nerves',
'3 Doors Down',
'ARXX',
'Crown Lands',
'The Undercover Dream Lovers',
'A Scent Like Wolves',
'Otis Junior',
'Madhav Gopi Nair',
'Advaita',
'John Campbell',
'Gymshorts',
'Liz Phair',
'Stephen Taranto',
'Deep Purple',
'Loyle Carner',
'Post Malone',
'Yehudit Ravitz',
'Olivia Rodrigo',
'Rema',
'Charlie Puth',
'Ravid Plotnik',
'Ulver',
'Alien Ant Farm',
'Live',
'Something Corporate',
'Buzzcocks',
'Enamour',
'mehro',
'Christian L√∂ffler',
'Rosie Flores',
'Ayron Jones',
'Dream Drop',
'Langston Hughes',
'Abby Wolfe',
'Eagles',
'Icon of Sin',
'Spirit Adrift',
'Shlomo Artzi',
'Eric B. & Rakim',
'Kayla Nicole, Taylor Girlz',
'The Rapture',
'A Day To Remember',
'Morgan Evans',
'Habalyanim',
'Wilco',
'DOECHII',
'Gabi Shoshan',
'Brothers Osborne',
'Zero 9:36',
'Róisín O',
'DIIV',
'The Swellers',
'Stalos & Oren Chen',
'Emery',
'Lionel Richie',
'Oren Barzilay',
'Foo Fighters',
'Gucci Mane',
'Luiz Bonfá, Panos Megarchiotis',
'Supergrass',
'The New Respects',
'Hadag Nahash',
'Julia Jacklin',
'Lazy Queen',
'No Somos Marineros',
'Keethan, Pavitra Krishnan, Shravan Sridhar',
'Acollective',
'No Memories of Tomorrow',
'The All-American Rejects',
'Nattali Rize',
'Apoorva Krishna',
'Ronnie Robinson Trio',
'Peled',
'Tay Money',
'Allegaeon',
'Quami',
'Oasis',
'Ghost Iris',
'Slowdive',
'Billie Eilish',
'Garbage',

'Britney Spears',
'Ruff Sqwad',
'Manal',
'The Black Keys',
'Noah Cyrus',
'Tomberlin',
'The Yardbirds',
'Wolfmother',
'Led Zeppelin',
'Hollow Front',
'The Beatles',
'Beloved',
'The Cult',
'Carach Angren',
'Fleshkiller',
"Edgar Winter's White Trash",
'Julia Kent',
'Arthur "Big Boy" Crudup',
'Big Brother & The Holding Company',
'Cody Johnson',
'Best Friends Club',
'Van Halen',
'Johnny Adams',
'Olivia Lunny',
'Merle Haggard',
'Blxst, Tyga, Ty Dolla $ign',
'MOSSS',
'The Sounds',
'Charlotte Day Wilson',
'Lower Than Atlantis',
'Jeris Johnson',
'Andi',
'Elton John',
'The Ben Webster Quintet',
'The Allergies',
'Shubha Mudgal',
'Galleons',
'Zara Larsson',
'Tori Harper',
'Saint Sister',
'ISLAND',
'Don Henley',
'Mukesh, Lata Mangeshkar',
'Ron Gallo',
'Thieves Like Us',
'Car, the garden',
'Sense Field',
'Earth, Wind & Fire',
'AJ Tracey',
'Type O Negative',
'Avril Lavigne',
'Tempa T',
'Viscera',
'Skating Polly',
'Samuel Lindon, Ensemble Cordes De La Mer',
'Black Pumas',
'The Jesus and Mary Chain',
'Charlie Feathers',
'Chicocurlyhead',
'Rochelle Jordan',
'Nasty Cherry',
'Power Trip',
'Nothing',
'Alien Sex Fiend',
'Lush',
'DJ Khaled',
'Remi Wolf',
'Data',
'Paladin',
'Michael Kiwanuka',
'Death Valley Girls',
'MONOEYES',
'Tuna',
'Concha Piquer',
'Ana Alcaide',
'TDJ',
'Miri Mesika',
'Institute',
'James Taylor',
'James Otto',
'elegant slims',
'Ms. Lauryn Hill',
'Goldfinger',
'Dire Straits',
'Bleachers',
'MAX RAD',
'The Faint',
'Kojaque',
'Limp Bizkit',

'Nick Hakim',
'Pritam, Arijit Singh',
'The Who',
'Caitlyn Smith',
'Leroy Carr',
'Fickle Friends',
'Trippie Redd',
'bl00dwave',
'Gaetano Donizetti, Academy of St. Martin in the Fields, Sir Neville Marriner',
'Chapterhouse',
'ILLENIUM',
'Kaiser Chiefs',
'ATB, Topic, A7S',
'MOSES',
'Shreya Ghoshal',
'Glass Animals',
'Aerosmith',
'Bathtub Cig',
'Louf',
'Yung Bleu',
'George Strait',
'DDG',
'Eugene Bridges',
'Dead Pony',
'iann dior',
'Idan Amedi',
'Lime Cordiale',
'Isaac DaBom',
'Ghetts',
'Shivam Pathak',
'The Soft Cavalry',
'Willie J Healey',
'The Kinks',
'Suldusk',
'Kurt Vile',
'Yameii Online',
'Jackie Hayes',
'Jack Fruit',
'Phoria',
'Shocking Blue',
'James Bay',
'Elliott Ok',
'The Zombies',
'AC/DC',
'Webb Pierce',
'Wallice',
'Tim Montana',
'Low (Acoustic)',
'Johnny Cash',
'LILHUDDY',
'The Wallflowers',
'Five Finger Death Punch',
'2 Chainz',
'Jess Locke',
'Wild Pink',
'Passion Pit',
'Shawnna',
'Walter Wolfman Washington',
'Jamestown Revival',
'Second to Sun',
'Armon',
'Long Distance Calling',
'The Jungle Giants',
'Sha Na Na',
'Beck',
'The Killers',
'Hemanth Bhagavath, Vignesh Prabhu, Swayam Prakash Prabhu',
'Parker Millsap',
'bbno$',
'No s√© a qui√©n matar',
'AWOLNATION',
'Bruce Springsteen',
"Jamie O'Neal",
'Braid',
'Rekha Bhardwaj',
'Tyla Yaweh',
'JunkBunny',
'Lisa Mann',
'Kylie V',
'Hazvuvim',
'The Walkmen',
'OK Go',
'Half the Animal',
'bloody/bath',
'Fontaines D.C.',
'Monolink',
'The Beaches',
'Dua Lipa',
'Suli Breaks',

'Malfunkshun',
'George Harrison',
'Mavi Phoenix',
'Jade Bird',
'King of Asgard',
'Touch√© Amor√©',
'Misplaced',
'Pote Baby',
'Run,ÄìD.M.C.',
'Journers',
'Miss May I',
'wave to earth',
'Fallen',
'Greta Van Fleet',
'Godsticks',
'Jan & Dean',
'The Pharcyde',
'Justus Bennetts',
'Salt-N-Pepa',
'Jake Milliner',
'21 Savage',
'George Moir',
'Amber Run',
'Whitney',
'P.O.D.',
'Monster Magnet',
'Press Club',
'The Futureheads',
'Gorgoroth',
'YONAKA',
'Fleetwood Mac',
'Arik Einstein',
'Pinkshinyultrablast',
'Migrant Motel',
'Teenage Wrist',
'Red Fang',
'The Devil and the Almighty Blues',
'Will Paquin',
'Pixies',
'Electric Light Orchestra',
'M√∂tley Cr√ºe',
'Lauv',
'Willie Dixon',
'Hollow',
'N.Flying',
'Life On Venus',
'The Robert Cray Band',
'Sammy Hagar',
'My Morning Jacket',
'Kansas',
'Rita Ora',
'Russell Dickerson',
'Dilated Peoples',
'Veil Of Maya',
'Kenny Neal',
"Look What You've Done",
'Fl√µstate',
'Florence + The Machine',
'Waterparks',
'YSN Flow',
'Bun B',
'TOOL',
'Yungeen Ace',
'The Standells',
'Miranda Lambert',
'Airbourne',
'The Fall',
'Haley Reinhart',
'The Last Poets',
'John Mayer',
'Sad Night Dynamite',
'Julian Lynch',
'Einherjer',
'Kris Barras Band',
'Honey Lung',
'The Alchemist',
'Rick Wakeman',
'Mondo Cozmo',
'Alluvial',
'Joe Nichols',
'Orions Belte',
'Sarit Hadad',
'The Darkness',
'Brittany Howard',
'Erik Ekholm',
'Sean Ono Lennon',
'Nick Cave & The Bad Seeds',
'FKA twigs',
'Glom',

'Antonín Dvořák, Busch Trio, Maria Milstein, Miguel Da Silva',
'Paul McCartney',
'Lata Mangeshkar',
'Sandeep Chowta, Abhay Nayampally, Alex Sill, Anthony Crawford',
'Hiatus Kaiyote',
'Yuval Dayan',
'Arlo Parks',
'Volumes',
'Jasta',
'Mudvayne',
'Mick Fleetwood and Friends',
'Westy',
'Gracie Abrams',
'Deap Vally',
'Hether',
'Confusing Paradise',
'Nothing But Thieves',
'LCY',
'KISS',
'Aya Zahavi Feiglin',
'Black Violin',
'IMMERSE',
'Ina Wroldsen',
'Itzik Kala',
'The Vines',
'Sentenced',
'Marianne Faithfull',
'Infinity Ripple',
'Lil Skies',
'Swamp Dogg',
'Savel',
'Rexx Life Raj',
'Kartell',
'The Bloody Beetroots',
'Trace Kotik',
'Ov Sulfur',
'black midi',
'Lauren Sanderson',
'my bloody valentine',
'Antonio Vivaldi, Raphael Wallfisch, City of London Sinfonia, Nicholas Kraemer',
'Joyner Lucas',
'Mabel',
'Ozzy Osbourne',
'Keith Jarrett Trio',
'St. Vincent',
'Michael Swissa',
'Good Health Good Wealth',
'Grip',
'Unknown Mortal Orchestra',
'FEVER 333',
'Billie Holiday',
'Motorjesus',
'Jubin Nautiyal, Gourov Dasgupta',
'Carlos Cano',
'Tui',
'Little Milton',
'Houston Stackhouse',
'Parkway Drive',
'Fab Samperi',
'The Pineapple Thief',
'Green Day',
'Shalom Gad',
'Palm Reader',
'Not Dead Yet',
'Kickin Valentina',
'Angel Bat Dawid',
'Oliver Tree',
'Stroke 9',
'Sports',
'Monta√±a',
'Imagine Dragons',
'The Cold Stares',
'Coal Chamber',
'Sam Hunt',
'Josh Turner',
'Wig Wam',
'Gary Moore',
'Blackstratblues',
'BROCKHAMPTON',
'gabriel black',
'David Keenan',
'Ikiatari',
'Mountain',
"Spock's Beard",
'Ghostemane',
'The Naked And Famous',
'Syberia',
'Yuval Banay',
'Meshuggah',

```
'Weezer',
'Carson Beyer',
'Future Islands',
'Citizen King',
'Tears For Fears',
'Imperio Argentina',
'Allman Brothers Band',
'Black Pistol Fire',
'Walking On Cars',
'Kills Birds',
"Clairy Browne & The Bangin' Rackettes",
'Frances Forever',
'Eric Von Schmidt',
'The Magnetic Fields',
'The Cash Box Kings',
'NanowaR of Steel',
'Walter Egan',
'Pretty Sick',
'Toby Keith',
'L.A. Guns',
'I Prevail',
'London Grammar',
'Temples',
'Rose Cousins',
'Semisonic',
'Betcha',
'Young Stoner Life',
'IOU',
'Ivan Bielsa, Timothy Reiger',
'Armlock',
'Atreyu',
'Benny Carter',
'Cedric Burnside',
'Electric Enemy',
'Seaway',
'Slash',
'Lloyd Cole and the Commotions',
'Nitin Sawhney',
'Years & Years',
'Ludacris',
'ONR',
'Kid Ink',
'Tommy Johnson',
"Guns N' Roses",
'The Dandy Warhols',
'Me And That Man',
'The Mayries',
'Yonatan Rozen',
'LPB Poody',
'VED',
'Jonas Blue',
'Tropidelic',
'Skindred',
'Crossfaith',
'N.O.R.E.',
'Christone "Kingfish" Ingram',
'NTMY',
'Rory Gallagher',
'Cheap Trick',
'Dispatch',
'Kina Grannis, Imaginary Future',
'Cosmo Sheldrake',
'Son Little',
'Pee Wee Crayton',
'Philip Sayce Group',
'MO3',
'Ryan Adams',
'Altin G√ºn',
'Bobby Verne',
'Good Thing',
'Peeping Drexels',
'Justin Timberlake',
'Beartooth',
'M-Beat',
'Moshe Peretz',
'The Strangeloves',
'Genesis Owusu',
'The Vaccines',
'Bonded',
'Don Rimini',
'Imperia',
'Shemekia Copeland',
'Passafire',
'Reese Quartet',
'serpentwithfeet',
'Briston Maroney',
'Transatlantic',
'Fleece',
'Paul Simon',
```

'Attila',
'Maudits',
'Laura Marling',
'Dan Mangan',
'Hermie',
'Kelsea Ballerini',
'Charli XCX',
'P!nk',
'Gwendolyn Brooks',
'West Thebarton',
'Arrested Youth',
'Trapdoor',
'Yung Mal',
'The Warning',
'Nicki Minaj',
'Sleater-Kinney',
'Skegss',
'Kinneret',
'No Vacation',
'Little Feat',
'Conquer Divide',
'Lucky Peterson',
'◊ò◊ú ◊ê◊ï◊®◊ù',
'Beggars & Thieves',
'After The Burial',
'Jake Bugg',
'Majestic, Boney M.',
'whenyoung',
'Shordie Shordie',
'Liquid Liquid',
'Long John Hunter',
'C. Ramchandra',
'Cake',
'The Pretty Reckless',
'Handbook',
'The Vaselines',
'Stellar',
'The Detroit Cobras',
'Prasanna',
'Vantage',
'Ministry',
'Lata Mangeshkar, S. P. Balasubrahmanyam',
'Benaia Barabi',
'Psychic Ills',
'Fuel',
'BOYTOY',
'Mike Shinoda',
'Quaker City Night Hawks',
'SK√ÅLD',
'Carla Geneve',
'Muse',
'Razorlight',
'Egypt Central',
'Civil Twilight',
'Ten Years After',
'Bon Jovi',
'Amir Ve Ben',
'TEMPOREX',
'Antônio Carlos Jobim, Sharon Isbin',
'Linkin Park',
'Nitty Gritty Dirt Band',
'The Kid LAROI, Polo G, Stunna Gambino',
'Andromida',
'Matthewdavid',
'Muddy Waters',
'Yemi Alade',
'The Wrecks',
'Danielle Bradbery',
'T. Rex',
'The Suitcase Junket',
'TWIN XL',
'A Perfect Circle',
'Lynyrd Skynyrd',
'Mazen',
'Rostam',
'Diamanda Gal√°s',
'Against The Current',
'The Undertones',
'Defamed',
'REO Speedwagon',
'grandson',
'Bruno Mars, Anderson .Paak, Silk Sonic',
'Claud',
'Kutiman',
'Tyler Leads',
'Flat Duo Jets',
'MARINA',
'Billie Marten',
'Starcastle',

'Doja Cat',
'Courtney Marie Andrews',
'Joey Gilmore',
'Gorillaz',
'Fox Stevenson',
'New Found Glory',
'Favourite People',
'Wiley',
'La Rata Bluesera',
'The National Parks',
'Suffer Like G Did',
'Major Murphy',
'The War and Treaty',
'Nascent',
'Marmozets',
'Metallica',
'Î∞ïÌòúÏßÑ Park Hye Jin',
'All That Remains',
'POLICE CAR COLLECTIVE',
'Gary Clark Jr.',
'Krokus',
'Slaves',
'Damien Burbank',
'Mc Rebecca',
'The Surfaris',
'Whitesnake',
'The Peterson Brothers',
'Powderfinger',
'Trivium',
'Bebe Rexha',
'Ed Sheeran',
'Grizzly Bear',
'Marcia Ball',
'Jerry Reed',
'Bombay Bicycle Club',
'Aviv Geffen',
'SawanoHiroyuki[nZk]:mizuki',
'Jean Charnaux',
'Tame Impala',
'Sea Girls',
'Blonde Redhead',
'Total Control',
'Midival Punditz, Kutle Khan',
'Godsmack',
'Idan Yaniv',
'Archetypes Collide',
'Far Away the Hills Are Green',
'Coldplay',
'Megan Thee Stallion',
'Praying Mantis',
'Lehakat Tzliley Haud',
'Jerry Garcia',
'Dudu Faruk',
'Dave Matthews Band',
'Las Penas',
'Zubi',
'LaSalle Street Trio',
'YSI',
'Saweetie',
'Valley',
'Ofdrykkja',
'Softcult',
'This Will Destroy You',
'Lucy Dacus',
'Metronomy',
'Bruce Hornsby',
'Duran Duran',
'Popp Hunna',
'Sizzy Rocket',
'Maximo Park',
'Dolly Parton',
'WALK THE MOON',
'IDLES',
'Slomosa',
'Kanye West',
'Aya Korem',
'Pepe Pecas',
'TORRES',
'Rajna Swaminathan',
'Def Leppard',
'Kevin Gates',
'Grateful Dead',
'Japandroids',
'18 Carat Affair',
'√ìlafur Arnalds',
'Elder Island',
'Kishore Kumar',
'Velvet Revolver',
'Lil Nas X',

'Henry Park',
'Kayzo',
'War',
'TheHxliday',
'Good Woman',
'Mayhem',
'Baseball Gregg',
'Marillion',
'Fredo Bang',
'Hein Talbot',
'Khruangbin',
'Tim McGraw',
'Leif Vollebekk',
'Miles Kane',
'Ratt',
'Jimmy Cliff',
'Django Django',
'The Goon Sax',
'Magic Bronson',
'courtship.',
'Frankie Ballard',
'Yehuda Poliker',
'Franz Ferdinand',
'Eve',
'Liily',
'Shea Diamond',
'Darlingside',
'Estrons',
'Doobie Bronson',
'Kishore Kumar, Amitabh Bachchan',
'DPR IAN',
'Spoon',
'pizzagirl',
'Holy Fawn',
'Ellie Goulding',
'Rhys Lewis',
'PENPALS',
'JUVENILE',
'LEN',
'Suicide',
'Depths of Hatred',
'Asha Bhosle, Mahendra Kapoor',
'Lawrence Rothman',
'Dead Meadow',
'Yuni Wa',
'Le Tigre',
'Yael Levi',
'Bush',
'Fred again..',
'Truffel the Phunky Phaqir',
'Big Spring',
'Audioslave',
'Powerwolf',
'Coma Cinema',
'John McLaughlin, Shankar Mahadevan, Zakir Hussain',
'The Police',
'Emil Landman',
'Sara Evans',
'Backstreet Boys',
'Kadinja',
'Sumo Cyco',
'Eloise',
'Primal Fear',
'Thelonious Monk Septet',
'Earth Trax',
'John Harvie',
'The Clash',
'The Easybeats',
'Activity',
'Olivia Meyer',
'John Hanks',
'bloodsimple',
'Tamikrest',
'David Bowie',
'Mandevilla',
'Carbine',
'daydream Masi',
'John Prine',
'Adriv°n Otero',
'Waylon Jennings',
'Suresh Wadkar, Bela',
'Lil Poppa',
'Jeff Beck',
'ILIRA',
'Eric Carmen',
'Sepultura',
'The Devil Wears Prada',
'Dikla',
'Billy Boy Arnold',

```
'Fly Right',
'Haricharan',
'NOROZ',
'The Adverts',
'Kula Shaker',
'Qrion',
'Proxy',
'Amitabh Bachchan, Pran',
'Sam Flax',
'Scorpions',
'Kishore Kumar, Alisha Chinai',
'Erick the Architect',
'Nasta Super',
'The Beths',
'Old Dominion',
'Big Audio Dynamite',
'Johnny "Guitar" Watson',
'D√òDSENGEL',
'Flying Lotus',
'Boy 8-Bit',
'Mazzy Star',
'Cracker',
'Billy Joel',
'Junkyard Drive',
'Jake Wesley Rogers',
'Hannah Wicklund',
'Middle Kids',
'Hatikva 6',
'Dylan Scott',
'Freak Slug',
'Reymour',
'Helena Deland',
'Eleine',
'Declan McKenna',
'Josh Teskey',
'Tones And I',
'Pavement',
'TV On The Radio',
'Ali Handal',
'Wetter',
'Giraffes? Giraffes!',
'Morray',
'–∏–Å–Ç–æ–á–Ω–∏–ʃ',
'Tesher',
'UniBe@t',
'blink-182',
'Mac Curtis',
'luxury elite',
'Down for Tomorrow',
'Mickey Lamantia',
'Samia',
'Udit Narayan, Sadhana Sargam',
'ZZ Top',
'Queensr√øche',
'Curley Weaver',
'Talking Heads',
'bonsi',
'SID',
'Prince Rapid',
'Aura Blum',
'WSTR',
'MC Lyte',
'Songhoy Blues',
'Dustin Tebbutt',
'The Smashing Pumpkins',
...]
```

In [74]:
```
len(df['Artist Name'].unique())
```

Out[74]: 8264

In [75]:
```
df=df.drop(['Artist Name','Track Name'],axis=1)
```

In [76]:
```
df
```

Out[76]:

	mode	time_signature	Class	Collabration	Popularity	danceability	energy	key	loudness	speechiness	acousticness	instrume
0	1	4	5	0	0.892192	1.862378	-0.441135	-1.551717	0.727199	-0.385493	-0.731701	-1.05(
1	1	4	10	0	0.546338	-0.971474	0.629952	-0.925698	0.158830	-0.478702	-0.783482	-6.62

2	1	4	6	0	-0.548867	-0.659270	-0.226918	0.013332	-0.118080	-0.338298	0.785790	-6.766
3	0	4	5	0	1.238046	1.856374	-0.299751	1.265371	0.334909	-0.302902	-0.718433	-1.050
4	1	4	10	0	0.488695	-2.262318	1.319732	-1.238708	0.899014	1.590773	-0.786495	-6.164
...	...	...	...	...	...	...	...	...	...	...	...	...
15608	0	3	6	0	-0.548867	-2.268322	-2.390514	0.326342	-2.316809	-0.470443	2.426584	2.440
15609	0	4	2	0	-1.010005	0.565530	-1.902098	1.578381	-0.579597	-0.569551	1.989686	-6.772
15610	0	4	8	0	-0.606509	0.085216	1.345439	-0.612688	0.797680	-0.117664	-0.786946	-6.768
15611	0	4	8	0	-0.894721	-1.974130	0.591393	0.013332	-1.227477	0.623288	-0.782867	2.789
15612	0	4	10	0	-0.087728	-0.863404	0.797042	-0.612688	0.637905	-0.260427	-0.767495	1.249

15613 rows × 16 columns

In [77]:
```
Y=df['Class']
Y
```

Out[77]:
```
0         5
1        10
2         6
3         5
4        10
         ..
15608     6
15609     2
15610     8
15611     8
15612    10
Name: Class, Length: 15613, dtype: int64
```

In [78]:
```
X=df.drop(['Class'],axis=1)
X
```

Out[78]:

	mode	time_signature	Collabration	Popularity	danceability	energy	key	loudness	speechiness	acousticness	instrumentalness
0	1	4	0	0.892192	1.862378	-0.441135	-1.551717	0.727199	-0.385493	-0.731701	-1.050342e-1
1	1	4	0	0.546338	-0.971474	0.629952	-0.925698	0.158830	-0.478702	-0.783482	-6.621742e-0
2	1	4	0	-0.548867	-0.659270	-0.226918	0.013332	-0.118080	-0.338298	0.785790	-6.766073e-0
3	0	4	0	1.238046	1.856374	-0.299751	1.265371	0.334909	-0.302902	-0.718433	-1.050342e-1
4	1	4	0	0.488695	-2.262318	1.319732	-1.238708	0.899014	1.590773	-0.786495	-6.164225e-0
...	...	...	...	...	...	...	...	...	...	...	...
15608	0	3	0	-0.548867	-2.268322	-2.390514	0.326342	-2.316809	-0.470443	2.426584	2.440876e+0
15609	0	4	0	-1.010005	0.565530	-1.902098	1.578381	-0.579597	-0.569551	1.989686	-6.772881e-0
15610	0	4	0	-0.606509	0.085216	1.345439	-0.612688	0.797680	-0.117664	-0.786946	-6.768344e-0
15611	0	4	0	-0.894721	-1.974130	0.591393	0.013332	-1.227477	0.623288	-0.782867	2.789028e+0
15612	0	4	0	-0.087728	-0.863404	0.797042	-0.612688	0.637905	-0.260427	-0.767495	1.249129e-0

15613 rows × 15 columns

In [79]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,train_size=0.8,random_state=77)
```

In [80]:
```
X_train
```

Out[80]:

	mode	time_signature	Collabration	Popularity	danceability	energy	key	loudness	speechiness	acousticness	instrumentalness
15356	0	4	0	0.315768	-0.214980	1.195486	1.265371	0.936637	0.138365	-0.741734	-1.050342e-1
8475	0	4	0	-0.491224	0.637578	0.042996	1.578381	0.340427	0.163142	-0.583803	-1.050342e-1
14954	0	4	0	-0.318297	0.211299	-0.132662	-1.238708	-0.178278	-0.632083	-0.521343	2.811733e+0
8125	1	4	1	-1.644071	0.019173	-0.659637	-1.238708	-1.056164	-0.604947	1.125600	1.320737e+0
8154	1	4	0	0.373411	0.229311	0.822748	-0.612688	0.983291	-0.486961	-0.786365	-2.043172e-0
...	...	...	...	...	...	...	...	...	...	...	...

2283	1	3	0	-0.087728	-0.929447	-2.133453	1.265371	-1.622777	-0.423248	2.167682	-3.817988e-0
10196	0	4	0	1.641542	0.937774	0.321479	-0.299678	0.272454	-0.112945	-0.553382	-6.020424e-0
11860	0	3	0	-0.721794	-1.637910	-0.394007	1.578381	-0.445155	-0.614386	-0.768530	2.989886e-0
8799	0	4	0	0.603980	0.529507	0.539981	0.013332	0.170368	-0.634443	-0.759436	-6.739810e-0
15063	1	4	0	1.295688	-1.043521	0.869876	-0.299678	1.448069	2.239697	-0.652736	-1.050342e-1

12490 rows × 15 columns

In [81]:
```
X_test
```

Out[81]:

	mode	time_signature	Collabration	Popularity	danceability	energy	key	loudness	speechiness	acousticness	instrumentalnes
15363	1	4	0	-0.721794	-0.419113	0.192949	1.265371	-0.085724	0.387316	1.093237	2.736048e+0
8994	1	4	0	-0.318297	-0.731317	-1.443673	-0.612688	-1.589919	0.611489	2.261534	-6.675478e-0
8095	1	4	0	-0.894721	-0.383090	-1.559350	-0.299678	-1.980704	-0.498759	0.462162	-6.769403e-0
1830	1	4	0	0.085199	0.751652	-0.235486	-1.238708	-0.165988	0.481705	-0.742381	-6.765619e-0
6435	1	4	0	0.488695	1.388068	-0.719618	-0.612688	0.171121	-0.468083	0.818153	-5.944738e-0
...	...	...	...	...	...	...	...	...	...	...	..
8822	0	4	0	1.295688	-0.317046	0.857023	0.013332	1.392135	0.257531	-0.246583	-1.050342e-1
988	0	4	0	-0.779436	-0.160944	0.432872	0.639352	0.783634	-0.453925	-0.786647	2.092725e+0
8417	1	4	0	0.085199	0.517499	-0.612509	-1.238708	-1.373708	-0.496399	-0.757235	-1.050342e-1
11540	1	3	0	1.526257	0.109232	-1.216602	0.013332	-1.908968	-0.599047	-0.324254	-2.081015e-0
12482	1	4	0	1.468615	-0.875412	1.126937	1.578381	0.276718	-0.232111	-0.784278	-2.194543e-0

3123 rows × 15 columns

In [82]:
```
Y_train
```

Out[82]:
```
15356    10
8475      9
14954    10
8125      0
8154      6
         ..
2283      7
10196    10
11860     8
8799     10
15063     5
Name: Class, Length: 12490, dtype: int64
```

In [83]:
```
Y_test
```

Out[83]:
```
15363     6
8994      9
8095      6
1830      6
6435      2
         ..
8822     10
988       6
8417     10
11540     6
12482    10
Name: Class, Length: 3123, dtype: int64
```

LOGISTIC REGRESSION

In [84]:
```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression(multi_class='multinomial')
lr.fit(X_train,Y_train)
```

Out[84]:
```
LogisticRegression(multi_class='multinomial')
```

In [85]:
```
lr.score(X_test,Y_test)
```

Out[85]:
```
0.48382965097662506
```

SVM - STANDARD VECTOR CLASSIFIER

In [86]:
```
from sklearn.svm import SVC
svc=SVC()
svc.fit(X_train,Y_train)
```

Out[86]:
```
SVC()
```

In [87]:
```
svc.score(X_test,Y_test)
```

Out[87]:
```
0.5084854306756325
```

RANDOM FOREST CLASSIFIER

In [88]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(X_train,Y_train)
```

Out[88]:
```
RandomForestClassifier()
```

In [89]:
```
rfc.score(X_test,Y_test)
```

Out[89]:
```
0.5004803073967339
```

K NEIGHBORS CLASSIFIER

In [90]:
```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier()
knn.fit(X_train,Y_train)
```

Out[90]:
```
KNeighborsClassifier()
```

In [91]:
```
knn.score(X_test,Y_test)
```

Out[91]:
```
0.4261927633685559
```

In [92]:
```
knn.predict(X_test)
```

Out[92]:
```
array([10,  9,  6, ..., 10, 10, 10], dtype=int64)
```