In [1]:
```
import pandas as pd
```

In [2]:
```
# Elimina les columnes del dataFrame passades per param

def drop_specific_cols(df, cols):
    df = df.drop(cols,axis=1)
    return df
```

In [3]:
```
# csv to df of only 10.000 first songs
msd_df = pd.read_csv('./datasets/msd_reduced.csv', nrows=36000, index_col=0)
```

In [4]:
```
# csv to df
grammy_df = pd.read_csv('./datasets/grammy_reduced.csv', index_col=0)
billboard_df = pd.read_csv('./datasets/billboard_reduced.csv', index_col=0)
spotify_df = pd.read_csv('./datasets/spotify_combined.csv', index_col=0)
```

In [5]:
```
# inner join between spotify and msd per obtenir els param fisics
new_df = pd.merge(msd_df, spotify_df, left_on=['title','artist_name'], right_on = ['title_spoty','all_artists'])
print ("Number of rows matched : %d"%new_df['popularity'].count())
new_df.describe()
```

Number of rows matched : 16251

Out[5]:

	danceability	duration	end_of_fade_in	energy	key	loudness	mode	start_of_fade_out	temp
count	16251.0	16251.000000	16251.000000	16251.0	16251.000000	16251.000000	16251.000000	16251.000000	16251.00000
mean	0.0	241.830750	0.813450	0.0	5.313396	-9.928168	0.679404	232.870017	124.71646
std	0.0	118.431612	1.993506	0.0	3.570588	5.008003	0.466720	116.066885	35.06637
min	0.0	0.626490	0.000000	0.0	0.000000	-47.403000	0.000000	0.626000	0.00000
25%	0.0	179.539140	0.000000	0.0	2.000000	-12.414000	0.000000	171.668500	98.49900
50%	0.0	225.488530	0.200000	0.0	5.000000	-8.904000	1.000000	216.317000	121.98800
75%	0.0	279.104850	0.438000	0.0	9.000000	-6.305000	1.000000	269.012000	145.66100
max	0.0	3029.080360	45.697000	0.0	11.000000	3.894000	1.000000	2999.583000	253.88700

In [6]:
```
# left joins for matches between spotify and grammy per obtenir el num de grammy's
new_df = pd.merge(new_df, grammy_df, how='left', left_on=['title','artist_name'], right_on = ['nominee','artist
# display matches
print ("Number of rows matched : %d"%new_df['nominee'].count())
new_df.describe()
```

Number of rows matched : 24

Out[6]:

	danceability	duration	end_of_fade_in	energy	key	loudness	mode	start_of_fade_out	temp
count	16251.0	16251.000000	16251.000000	16251.0	16251.000000	16251.000000	16251.000000	16251.000000	16251.00000
mean	0.0	241.830750	0.813450	0.0	5.313396	-9.928168	0.679404	232.870017	124.71646
std	0.0	118.431612	1.993506	0.0	3.570588	5.008003	0.466720	116.066885	35.06637
min	0.0	0.626490	0.000000	0.0	0.000000	-47.403000	0.000000	0.626000	0.00000
25%	0.0	179.539140	0.000000	0.0	2.000000	-12.414000	0.000000	171.668500	98.49900
50%	0.0	225.488530	0.200000	0.0	5.000000	-8.904000	1.000000	216.317000	121.98800
75%	0.0	279.104850	0.438000	0.0	9.000000	-6.305000	1.000000	269.012000	145.66100
max	0.0	3029.080360	45.697000	0.0	11.000000	3.894000	1.000000	2999.583000	253.88700

In [7]:
```
new_df = pd.merge(new_df, billboard_df, how='left', left_on=['title','artist_name'], right_on = ['song','artist
print ("Number of rows matched : %d"%new_df['song'].count())
new_df.describe()
```

Number of rows matched : 321

Out[7]:

	danceability	duration	end_of_fade_in	energy	key	loudness	mode	start_of_fade_out	temp
count	16251.0	16251.000000	16251.000000	16251.0	16251.000000	16251.000000	16251.000000	16251.000000	16251.00000
mean	0.0	241.830750	0.813450	0.0	5.313396	-9.928168	0.679404	232.870017	124.71646
std	0.0	118.431612	1.993506	0.0	3.570588	5.008003	0.466720	116.066885	35.06637
min	0.0	0.626490	0.000000	0.0	0.000000	-47.403000	0.000000	0.626000	0.00000
25%	0.0	179.539140	0.000000	0.0	2.000000	-12.414000	0.000000	171.668500	98.49900
50%	0.0	225.488530	0.200000	0.0	5.000000	-8.904000	1.000000	216.317000	121.98800
75%	0.0	279.104850	0.438000	0.0	9.000000	-6.305000	1.000000	269.012000	145.66100
max	0.0	3029.080360	45.697000	0.0	11.000000	3.894000	1.000000	2999.583000	253.88700

8 rows × 22 columns

In [8]:
```
display(new_df)
```

	danceability	duration	end_of_fade_in	energy	key	loudness	mode	start_of_fade_out	tempo	time_signature	...	popul
0	0.0	138.97098	0.000	0.0	7	-2.060	1	138.971	177.768	4	...	
1	0.0	221.20444	0.165	0.0	11	-12.214	0	212.120	98.020	4	...	
2	0.0	318.45832	0.502	0.0	10	-10.670	1	306.265	67.567	3	...	
3	0.0	262.26893	0.194	0.0	11	-3.925	1	259.419	122.332	4	...	
4	0.0	196.02240	0.000	0.0	8	-6.366	1	185.202	189.346	7	...	
...	...	...	...	...	...	...	...	...	...	...	...	
16246	0.0	196.67546	0.000	0.0	0	-5.117	1	189.098	123.084	3	...	
16247	0.0	268.53832	3.959	0.0	0	-8.746	1	253.951	118.721	5	...	
16248	0.0	208.45669	0.000	0.0	9	-15.338	1	190.537	120.720	4	...	
16249	0.0	228.17914	0.293	0.0	11	-6.728	1	223.742	130.931	1	...	
16250	0.0	98.76853	0.490	0.0	11	-7.962	1	98.769	196.265	1	...	

16251 rows × 31 columns

In [9]:
```
# Creem una nova columna, e_grammy, que pren valors binaris 0 o 1 segons si la cançó ha guanyat un Grammy.
# Ojo perquè una cançó pot guanyar més d'un Grammy.
new_df['e_grammy'] = new_df['nominee'].notnull().astype('int')
```

In [10]:
```
new_df.describe()
```

Out[10]:

	danceability	duration	end_of_fade_in	energy	key	loudness	mode	start_of_fade_out	temp
count	16251.0	16251.000000	16251.000000	16251.0	16251.000000	16251.000000	16251.000000	16251.000000	16251.00000
mean	0.0	241.830750	0.813450	0.0	5.313396	-9.928168	0.679404	232.870017	124.71646
std	0.0	118.431612	1.993506	0.0	3.570588	5.008003	0.466720	116.066885	35.06637
min	0.0	0.626490	0.000000	0.0	0.000000	-47.403000	0.000000	0.626000	0.00000
25%	0.0	179.539140	0.000000	0.0	2.000000	-12.414000	0.000000	171.668500	98.49900
50%	0.0	225.488530	0.200000	0.0	5.000000	-8.904000	1.000000	216.317000	121.98800
75%	0.0	279.104850	0.438000	0.0	9.000000	-6.305000	1.000000	269.012000	145.66100
max	0.0	3029.080360	45.697000	0.0	11.000000	3.894000	1.000000	2999.583000	253.88700

8 rows × 23 columns

In [11]:
```
count = (new_df['e_grammy'] != 0).sum()
print(count)
```

24

In [12]:
```
#yay!
```

In [13]:
```
new_df['is_billboard'] = new_df['peak-rank'].notnull().astype('int')
```

In [14]:
```
def billboard_success(m, s, is_billboard):
    if is_billboard:
        return (101 - m + s)
    else:
        return 0
```

In [15]:
```
# Per cada fila de la taula new_df, cridem la funció billboard_success amb els atributs neg. i guardem el valor
new_df['is_billboard'] = new_df.apply(lambda x: billboard_success(x['peak-rank'], x['weeks-on-board'],x['is_bill
```

In [16]:
```
display(new_df)
```

	danceability	duration	end_of_fade_in	energy	key	loudness	mode	start_of_fade_out	tempo	time_signature	...	artist_
0	0.0	138.97098	0.000	0.0	7	-2.060	1	138.971	177.768	4	...	Na
1	0.0	221.20444	0.165	0.0	11	-12.214	0	212.120	98.020	4	...	Na
2	0.0	318.45832	0.502	0.0	10	-10.670	1	306.265	67.567	3	...	Na
3	0.0	262.26893	0.194	0.0	11	-3.925	1	259.419	122.332	4	...	Na
4	0.0	196.02240	0.000	0.0	8	-6.366	1	185.202	189.346	7	...	Na
...	...	...	...	...	...	...	...	...	...	...	...	
16246	0.0	196.67546	0.000	0.0	0	-5.117	1	189.098	123.084	3	...	Na
16247	0.0	268.53832	3.959	0.0	0	-8.746	1	253.951	118.721	5	...	Na
16248	0.0	208.45669	0.000	0.0	9	-15.338	1	190.537	120.720	4	...	Na
16249	0.0	228.17914	0.293	0.0	11	-6.728	1	223.742	130.931	1	...	Na
16250	0.0	98.76853	0.490	0.0	11	-7.962	1	98.769	196.265	1	...	Na

16251 rows × 34 columns

In [17]:
```
new_df.describe()
```

Out[17]:

	danceability	duration	end_of_fade_in	energy	key	loudness	mode	start_of_fade_out	temp
count	16251.0	16251.000000	16251.000000	16251.0	16251.000000	16251.000000	16251.000000	16251.000000	16251.00000
mean	0.0	241.830750	0.813450	0.0	5.313396	-9.928168	0.679404	232.870017	124.71646
std	0.0	118.431612	1.993506	0.0	3.570588	5.008003	0.466720	116.066885	35.06637
min	0.0	0.626490	0.000000	0.0	0.000000	-47.403000	0.000000	0.626000	0.00000
25%	0.0	179.539140	0.000000	0.0	2.000000	-12.414000	0.000000	171.668500	98.49900
50%	0.0	225.488530	0.200000	0.0	5.000000	-8.904000	1.000000	216.317000	121.98800
75%	0.0	279.104850	0.438000	0.0	9.000000	-6.305000	1.000000	269.012000	145.66100
max	0.0	3029.080360	45.697000	0.0	11.000000	3.894000	1.000000	2999.583000	253.88700

8 rows × 25 columns

In [18]:
```
count_b = (new_df['e_billboard'] != 0).sum()
print(count_b)
```

321

In [19]:
```
new_df['popularity'] = new_df['popularity'].fillna(0)
new_df['e_spotify'] = 0
new_df.loc[new_df['popularity'] >= new_df['popularity'].quantile(0.75), 'e_spotify'] = 1
new_df.loc[new_df['popularity'] < new_df['popularity'].quantile(0.75), 'e_spotify'] = 0
```

In [20]:
```
count_s = (new_df['e_spotify'] != 0).sum()
print(count_s)
```

4146

In [21]:
```
# Eliminem columnes innecessàries per a estudiar les dades
cols = [ 'nominee', 'artist_x', 'rank', 'song', 'artist_y', 'peak-rank', 'weeks-on-board', 'is_billboard', 'all
new_df = drop_specific_cols(new_df, cols)
```

In [22]:
```
display(new_df)
```

	danceability	duration	end_of_fade_in	energy	key	loudness	mode	start_of_fade_out	tempo	time_signature	...	artist_
0	0.0	138.97098	0.000	0.0	7	-2.060	1	138.971	177.768	4	...	
1	0.0	221.20444	0.165	0.0	11	-12.214	0	212.120	98.020	4	...	
2	0.0	318.45832	0.502	0.0	10	-10.670	1	306.265	67.567	3	...	
3	0.0	262.26893	0.194	0.0	11	-3.925	1	259.419	122.332	4	...	
4	0.0	196.02240	0.000	0.0	8	-6.366	1	185.202	189.346	7	...	
...	...	...	...	...	...	...	...	...	...	...	...	
16246	0.0	196.67546	0.000	0.0	0	-5.117	1	189.098	123.084	3	...	
16247	0.0	268.53832	3.959	0.0	0	-8.746	1	253.951	118.721	5	...	
16248	0.0	208.45669	0.000	0.0	9	-15.338	1	190.537	120.720	4	...	
16249	0.0	228.17914	0.293	0.0	11	-6.728	1	223.742	130.931	1	...	
16250	0.0	98.76853	0.490	0.0	11	-7.962	1	98.769	196.265	1	...	Santa

16251 rows × 23 columns

In [23]:
```
new_df.describe()
```

Out[23]:

	danceability	duration	end_of_fade_in	energy	key	loudness	mode	start_of_fade_out	temp
count	16251.0	16251.000000	16251.000000	16251.0	16251.000000	16251.000000	16251.000000	16251.000000	16251.00000
mean	0.0	241.830750	0.813450	0.0	5.313396	-9.928168	0.679404	232.870017	124.71646
std	0.0	118.431612	1.993506	0.0	3.570588	5.008003	0.466720	116.066885	35.06637
min	0.0	0.626490	0.000000	0.0	0.000000	-47.403000	0.000000	0.626000	0.00000
25%	0.0	179.539140	0.000000	0.0	2.000000	-12.414000	0.000000	171.668500	98.49900
50%	0.0	225.488530	0.200000	0.0	5.000000	-8.904000	1.000000	216.317000	121.98800
75%	0.0	279.104850	0.438000	0.0	9.000000	-6.305000	1.000000	269.012000	145.66100
max	0.0	3029.080360	45.697000	0.0	11.000000	3.894000	1.000000	2999.583000	253.88700

In [24]:
```
# export to csv file
new_df.to_csv("successes55000_with_spotify.csv")
```

In [ ]:
```

```