Netegem tots els datasets

Realitzem una neteja de les files i columnes dels datasets MillionSongs, Billboard i Grammy per a reduir-ne la mida i evitar claus primàries repetides.

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

Million Songs Dataset

In [2]:
```
"""
Eliminar files seleccionades
"""
def clean(df, keep_rows):
    # replace NaN values with 0
    df = df.fillna(0)
    #Drop rows with category = 2
    df = df.loc[df['category'].isin(keep_rows)]
```

In [3]:
```
"""
Eliminar columnes seleccionades
"""
def drop_specific_cols(df, cols):
    df = df.drop(cols,axis=1)
    return df
```

In [4]:
```
def get_songs_df_from_csv(filename):
    return pd.read_csv(filename,index_col=0)
```

In [5]:
```
#GETTING DATA
songs_df = get_songs_df_from_csv('./datasets/million_songs_dataset.csv')

#During the csv making process, I was merging different dfs into one df and df index info was lost. Let's reset
songs_df.reset_index(drop=True,inplace=True)

print ("Number of Rows: %d"%len(songs_df))
print ("Number of Columns: %d"%len(songs_df.columns))
pd.set_option('display.max_columns', None)
display(songs_df)
```

```
Number of Rows: 1000000
Number of Columns: 53
```

	analysis_sample_rate	audio_md5	danceability	duration	end_of_fade_in	energy	idx_bars_confic
0	22050	aee982091178fc734e7694c5432990ca	0.0	252.05506	2.049	0.0	
1	22050	ed222d07c83bac7689d52753610a513a	0.0	156.55138	0.258	0.0	
2	22050	96c7104889a128fef84fa469d60e380c	0.0	138.97098	0.000	0.0	
3	22050	0f7da84b6b583e3846c7e022fb3a92a2	0.0	145.05751	0.000	0.0	
4	22050	228ad6392ad8001b02811533f34c72fd	0.0	514.29832	0.000	0.0	
...	...	...	...	...	...	...	...
999995	22050	31c447839cdc20465e03ae6a05883143	0.0	217.44281		0.0	
999996	22050	e30bcbd29572ac7d085acd5b26a97464	0.0	244.16608	3.048	0.0	
999997	22050	7d0656b833e183244a3c3ed023fcbb70a	0.0	553.03791	0.223	0.0	
999998	22050	32473a8e2d20f3efbdcb3caa57d4bf35	0.0	484.51873	0.595	0.0	
999999	22050	7c4a9f610c8f736467a1463027a8bc40	0.0	295.07873	0.000	0.0	

1000000 rows × 53 columns

In [6]:
```
#DATA CLEANING

# per eliminar files
# df = clean(songs_df)

songs_df = songs_df.drop_duplicates(subset=['title', 'artist_name'], keep='first')

# per eliminar columnes
cols = ['analysis_sample_rate','audio_md5','idx_bars_confidence','idx_bars_start','idx_beats_confidence','idx_b
df = drop_specific_cols(songs_df, cols)

# display
print ("Number of Columns after data cleansing: %d"%len(df.columns))
print ("Number of rows after cleansing: %d"%df.danceability.count())
df.head()
```

```
Number of Columns after data cleansing: 20
Number of rows after cleansing: 924096
```

Out[6]:

	danceability	duration	end_of_fade_in	energy	key	loudness	mode	start_of_fade_out	tempo	time_signature	artist_familiarit
0	0.0	252.05506	2.049	0.0	10	-4.829	0	236.635	87.002	4	0.64982
1	0.0	156.55138	0.258	0.0	9	-10.555	1	148.660	150.778	1	0.43966
2	0.0	138.97098	0.000	0.0	7	-2.060	1	138.971	177.768	4	0.64366
3	0.0	145.05751	0.000	0.0	7	-4.654	1	138.687	87.433	4	0.44850
4	0.0	514.29832	0.000	0.0	5	-7.806	0	506.717	140.035	4	0.00000

In [7]:
```
# export to csv file
df.to_csv("bad_reduced.csv")
```

Grammy

Categories grammy a considerar: res que sigui performance, album Record Of The Year Song Of The Year Best Dance/Electronic Recording Best Rock Song Best R&B Song Best Rap Song Best Country Song Best Improvised Jazz Solo Best Gospel Performance/Song Best Gospel Song Best Contemporary Christian Performance/Song Best Contemporary Christian Song Best American Roots Song Best Contemporary Song Best Disco Recording Best Ethnic Or Traditional Folk Recording

In [8]:
```
#GETTING DATA
grammy_df = get_songs_df_from_csv('./datasets/grammy.csv')

#During the csv making process, I was merging different dfs into one df and df index info was lost. Let's reset
grammy_df.reset_index(drop=True,inplace=True)

print ("Number of Rows: %d"%len(grammy_df))
print ("Number of Columns: %d"%len(grammy_df.columns))
pd.set_option('display.max_columns', None)
display(grammy_df)
```

```
Number of Rows: 4810
Number of Columns: 9
```

	title	published_at	updated_at	category	nominee	artist	workers	i
0	62nd Annual GRAMMY Awards (2019)	2020-05-19T00:10:26-07:00	2020-05-19T00:10:26-07:00	Record Of The Year	Bad Guy	Billie Eilish	Finneas O'Connell, producer; Rob Kinelski & Fi...	https://www.grammy.com/sites/com/files/styles
1	62nd Annual GRAMMY Awards (2019)	2020-05-19T00:10:26-07:00	2020-05-19T00:10:26-07:00	Record Of The Year	Hey, Ma	Bon Iver	BJ Burton, Brad Cook, Chris Messina & Justin V...	https://www.grammy.com/sites/com/files/styles
2	62nd Annual GRAMMY Awards (2019)	2020-05-19T00:10:26-07:00	2020-05-19T00:10:26-07:00	Record Of The Year	7 rings	Ariana Grande	Charles Anderson, Tommy Brown, Michael Foster ...	https://www.grammy.com/sites/com/files/styles
3	62nd Annual GRAMMY Awards (2019)	2020-05-19T00:10:26-07:00	2020-05-19T00:10:26-07:00	Record Of The Year	Hard Place	H.E.R.	Rodney "Darkchild" Jerkins, producer; Joseph H...	https://www.grammy.com/sites/com/files/styles
4	62nd Annual GRAMMY Awards (2019)	2020-05-19T00:10:26-07:00	2020-05-19T00:10:26-07:00	Record Of The Year	Talk	Khalid	Disclosure & Denis Kosiak, producers; Ingmar C...	https://www.grammy.com/sites/com/files/styles
...	...	...	...	...	...	...	...	
4805	1st Annual GRAMMY Awards (1958)	2017-11-28T00:03:45-08:00	2019-09-10T01:11:09-07:00	Best Classical Performance - Instrumentalist f...	Tchaikovsky: Piano Concerto No. 1 In B Flat Mi...	NaN	Van Cliburn, artist (Symphony Of The Air Orche...	h
4806	1st Annual GRAMMY Awards (1958)	2017-11-28T00:03:45-08:00	2019-09-10T01:11:09-07:00	Best Classical Performance - Instrumentalist f...	Segovia Golden Jubilee	NaN	Andres Segovia, artist	https://www.grammy.com/sites/com/files/styles
4807	1st Annual GRAMMY Awards (1958)	2017-11-28T00:03:45-08:00	2019-09-10T01:11:09-07:00	Best Classical Performance - Chamber Music (In...	Beethoven: Quartet 130	NaN	Hollywood String Quartet (Alvin Dinkin, Paul S...	h
4808	1st Annual GRAMMY Awards (1958)	2017-11-28T00:03:45-08:00	2019-09-10T01:11:09-07:00	Best Classical Performance - Vocal Soloist (Wi...	Operatic Recital	NaN	NaN	h
4809	1st Annual GRAMMY Awards (1958)	2017-11-28T00:03:45-08:00	2019-09-10T01:11:09-07:00	Best Classical Performance - Operatic Or Choral	Virtuoso	NaN	Roger Wagner, choir director	h

4810 rows × 9 columns

In [9]:
```
#DATA CLEANING

# per eliminar columnes
cols = ['title', 'published_at', 'updated_at', 'img', 'winner']
grammy_df = drop_specific_cols(grammy_df, cols)

# per eliminar files
keep_rows = ['Record Of The Year', 'Song Of The Year', 'Best Dance/Electronic Recording', 'Best Rock Song', 'Be
df = clean(grammy_df, keep_rows)

# display
print ("Number of Columns after data cleansing: %d"%len(grammy_df.columns))
print ("Number of rows after cleansing: %d"%grammy_df.nominee.count())
grammy_df.head()
```

```
Number of Columns after data cleansing: 4
Number of rows after cleansing: 4804
```

Out[9]:

	category	nominee	artist	workers
0	Record Of The Year	Bad Guy	Billie Eilish	Finneas O'Connell, producer; Rob Kinelski & Fi...
1	Record Of The Year	Hey, Ma	Bon Iver	BJ Burton, Brad Cook, Chris Messina & Justin V...
2	Record Of The Year	7 rings	Ariana Grande	Charles Anderson, Tommy Brown, Michael Foster ...
3	Record Of The Year	Hard Place	H.E.R.	Rodney "Darkchild" Jerkins, producer; Joseph H...
4	Record Of The Year	Talk	Khalid	Disclosure & Denis Kosiak, producers; Ingmar C...

In [10]:
```
# per eliminar keys repetides, poden creer nova columna que sigui num. premis de la cançó
grammy_df = grammy_df.groupby(['nominee', 'artist']).size().reset_index(name='prizes')
```

In [11]:
```
# display
print ("Number of rows after grouping: %d"%grammy_df.nominee.count())
grammy_df.head()
```

```
Number of rows after grouping: 2841
```

Out[11]:

	nominee	artist	prizes
0	#Edisco	Alejandro Sanz	1
1	'Round Midnight	Bobby McFerrin	2
2	'Til Summer Comes Around	Keith Urban	1
3	(Everything I Do) I Do It For You (From Robin ...	Bryan Adams, Michael Kamen & Robert John "Mutt...	1
4	(I'm A) Stand By My Woman Man	Ronnie Milsap	1

In [12]:
```
# export to csv file
grammy_df.to_csv("grammy_reduced.csv")
```

Billboard

In [13]:
```
#GETTING DATA
billboard_df = get_songs_df_from_csv('./datasets/billboard.csv')

#During the csv making process, I was merging different dfs into one df and df index info was lost. Let's reset
billboard_df.reset_index(drop=True,inplace=True)

print ("Number of Rows: %d"%len(billboard_df))
print ("Number of Columns: %d"%len(billboard_df.columns))
pd.set_option('display.max_columns', None)
display(billboard_df)
```

```
Number of Rows: 330087
Number of Columns: 6
```

	rank	song	artist	last-week	peak-rank	weeks-on-board
0	1	Easy On Me	Adele	1.0	1	3
1	2	Stay	The Kid LAROI & Justin Bieber	2.0	1	16
2	3	Industry Baby	Lil Nas X & Jack Harlow	3.0	1	14
3	4	Fancy Like	Walker Hayes	4.0	3	19
4	5	Bad Habits	Ed Sheeran	5.0	2	18
...	...	...	...	...	...	...
330082	96	Over And Over	Thurston Harris	NaN	96	1
330083	97	I Believe In You	Robert & Johnny	NaN	97	1
330084	98	Little Serenade	The Ames Brothers	NaN	98	1
330085	99	I'll Get By (As Long As I Have You)	Billy Williams	NaN	99	1
330086	100	Judy	Frankie Vaughan	NaN	100	1

330087 rows × 6 columns

In [14]:
```
#DATA CLEANING



billboard_df.drop_duplicates(subset = ['song', 'artist'], keep = 'first', inplace = True)


# per eliminar columnes
cols = ['last-week']
billboard_df = drop_specific_cols(billboard_df, cols)


# display
print ("Number of Columns after data cleansing: %d"%len(billboard_df.columns))
print ("Number of rows after cleansing: %d"%billboard_df.song.count())
billboard_df.head()
```

```
Number of Columns after data cleansing: 5
Number of rows after cleansing: 29681
```

Out[14]:

	rank	song	artist	peak-rank	weeks-on-board
0	1	Easy On Me	Adele	1	3
1	2	Stay	The Kid LAROI & Justin Bieber	1	16
2	3	Industry Baby	Lil Nas X & Jack Harlow	1	14
3	4	Fancy Like	Walker Hayes	3	19
4	5	Bad Habits	Ed Sheeran	2	18

In [15]:
```
# export to csv file
billboard_df.to_csv("billboard_reduced.csv")
```