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 # csv to df of only 10.000 first songs In [3]: 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 físics 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 duration end_of_fade_in energy Out[5]: danceability **loudness** mode start_of_fade_out key temp 16251.0 16251.000000 16251.000000 16251.0 16251.000000 16251.000000 16251.000000 16251.000000 16251.00000 count mean 0.0 241.830750 0.813450 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 0.00000 min 0.0 0.626490 0.000000 0.0 0.000000 -47.403000 0.000000 0.626000 25% 0.0 179.539140 0.000000 0.0 2.000000 -12.414000 0.000000 171.668500 98.49900 50% 225.488530 0.200000 5.000000 -8.904000 1.000000 216.317000 121.98800 0.0 75% 0.0 279.104850 0.438000 0.0 9.000000 -6.305000 1.000000 269.012000 145.66100 3029.080360 11.000000 253.88700 0.0 45.697000 0.0 3.894000 1.000000 2999.583000 max 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 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 16251.0 16251.000000 16251.000000 16251.0 16251.000000 16251.000000 16251.000000 16251.000000 16251.00000 count mean 0.0 241.830750 0.813450 0.0 5.313396 -9.928168 0.679404 232.870017 124.71646 0.0 118.431612 1.993506 0.0 3.570588 5.008003 0.466720 116.066885 35.06637 std 0.626490 0.000000 0.000000 -47.403000 0.000000 0.626000 0.00000 0.0 0.0 min 25% 0.0 179.539140 0.000000 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% 279.104850 9.000000 -6.305000 269.012000 145.66100 0.0 0.438000 0.0 1.000000 0.0 3029.080360 45.697000 0.0 11.000000 3.894000 1.000000 2999.583000 253.88700 max 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]: duration end_of_fade_in energy danceability key loudness mode start_of_fade_out temp 16251.0 16251.000000 16251.000000 16251.00000 16251.000000 16251.0 16251.000000 16251.000000 16251.000000 count 0.0 241.830750 0.813450 0.0 5.313396 -9.928168 0.679404 232.870017 124.71646 mean 0.0 118.431612 1.993506 3.570588 5.008003 116.066885 35.06637 std 0.0 0.466720 0.000000 min 0.0 0.626490 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 3029.080360 45.697000 2999.583000 max 0.0 0.0 11.000000 3.894000 1.000000 253.88700 8 rows × 22 columns display(new df) In [8]: danceability tempo time_signature ... popula duration end_of_fade_in energy key loudness mode start_of_fade_out 0.0 138.97098 0.000 0.0 7 -2.060 1 138.971 177.768 1 0.0 221.20444 0.165 0.0 11 -12.214 0 212.120 98.020 4 2 0.502 10 306.265 0.0 318.45832 0.0 -10.670 1 67.567 3 3 0.0 262.26893 0.194 0.0 -3.925 4 11 1 259.419 122.332 4 0.0 196.02240 0.000 8 -6.366 1 185.202 7 ... 0.0 189.346 • • • • • • 189.098 123.084 3 ... 16246 0.0 196.67546 0.000 0 0.0 -5.117 1 16247 0.0 268.53832 3.959 0.0 0 -8.746 253.951 118.721 5 16248 208.45669 0.000 9 1 190.537 120.720 0.0 0.0 -15.338 4 ... 16249 0.0 228.17914 0.293 0.0 11 -6.728 223.742 130.931 16250 0.0 98.76853 0.490 0.0 11 -7.962 1 98.769 196.265 1 ... $16251 \text{ rows} \times 31 \text{ columns}$ # Creem una nova columna, e_grammy, que pren valors binaris 0 o 1 segons si la cançó ha guanyat un Grammy. In [9]: # Ojo perquè una cançó pot guanyar més d'un Grammy. new df['e grammy'] = new df['nominee'].notnull().astype('int') new_df.describe() In [10]: Out[10]: danceability duration end_of_fade_in energy key **loudness** start_of_fade_out temp count 16251.0 16251.000000 16251.000000 16251.0 16251.000000 16251.000000 16251.000000 16251.000000 16251.00000 0.0 241.830750 0.813450 0.0 5.313396 -9.928168 0.679404 232.870017 124.71646 mean 1.993506 118.431612 3.570588 5.008003 116.066885 35.06637 std 0.0 0.0 0.466720 -47.403000 0.00000 0.000000 0.000000 0.626000 min 0.0 0.626490 0.000000 25% 0.0 179.539140 0.000000 0.0 2.000000 -12.414000 0.000000 171.668500 98.49900 50% 0.0 0.200000 0.0 5.000000 -8.904000 1.000000 216.317000 121.98800 225.488530 9.000000 269.012000 75% 0.0 279.104850 0.438000 0.0 -6.305000 1.000000 145.66100 11.000000 3029.080360 45.697000 3.894000 1.000000 2999.583000 253.88700 max 0.0 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 seg. i guardem el valor new df['e billboard'] = new df.apply(lambda x: billboard success(x['peak-rank'], x['weeks-on-board'],x['is bill In [16]: display(new_df) duration end_of_fade_in energy key loudness mode start_of_fade_out tempo time signature ... artist danceability 4 ... 0 0.0 138.97098 0.000 0.0 7 -2.060 1 138.971 177.768 Νέ 1 0.0 221.20444 11 -12.2140 98.020 0.165 0.0 212.120 4 ... Νá 2 318.45832 0.502 3 ... 0.0 0.0 10 -10.670 1 306.265 67.567 Νá 3 0.0 262.26893 0.194 0.0 11 -3.925 259.419 122.332 4 ... Νá 189.346 4 0.0 196.02240 0.000 0.0 8 -6.366 1 185.202 7 ... Νá • • • • • • 189.098 3 ... 16246 0.0 196.67546 0.000 0.0 0 -5.117 1 123.084 Νá 5 ... 16247 0.0 268.53832 3.959 0.0 0 -8.746 253.951 118.721 Νá 16248 0.0 208.45669 0.000 0.0 9 -15.338 1 190.537 120.720 4 ... 0.0 223,742 1 ... 16249 228.17914 0.293 0.0 11 -6.728 130.931 Na 16250 0.0 0.490 0.0 11 -7.962 1 1 ... 98.76853 98.769 196.265 Νέ 16251 rows × 34 columns new_df.describe() In [17]: Out[17]: danceability duration end_of_fade_in energy **loudness** mode start_of_fade_out key temp count 16251.0 16251.000000 16251.000000 16251.0 16251.000000 16251.000000 16251.000000 16251.000000 16251.00000 -9.928168 0.0 241.830750 0.813450 0.0 5.313396 232.870017 mean 0.679404 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% 0.200000 5.000000 -8.904000 0.0 225.488530 0.0 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 0.0 3029.080360 45.697000 0.0 11.000000 3.894000 1.000000 2999.583000 253.88700 max 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</pre> count_s = (new_df['e_spotify'] != 0).sum() In [20]: print(count_s) 4146 # Eliminem columnes innecessàries per a estudiar les dades In [21]: cols = ['nominee', 'artist_x', 'rank', 'song', 'artist_y', 'peak-rank', 'weeks-on-board', 'is_billboard', 'all new_df = drop_specific_cols(new_df, cols) display(new_df) In [22]: danceability tempo time_signature ... artist_ duration end_of_fade_in energy key loudness mode start_of_fade_out 0 0.0 138.97098 0.000 0.0 -2.060 7 1 138.971 177.768 0.0 221.20444 0.165 0.0 11 -12.214 212.120 98.020 3 ... 2 0.0 0.502 10 306.265 318.45832 0.0 -10.670 1 67.567 0.0 262.26893 0.194 0.0 11 -3.925 259.419 122.332 1 4 0.0 196.02240 0.000 0.0 8 -6.3661 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 -15.338 0.0 208.45669 0.000 0.0 9 1 190.537 120.720 4 228.17914 16249 0.0 0.293 0.0 11 223.742 130.931 -6.7281 1 ... Santa 16250 0.0 98.76853 0.490 0.0 11 -7.9621 98.769 196.265 16251 rows × 23 columns new_df.describe() In [23]: Out[23]: danceability duration end_of_fade_in loudness start_of_fade_out energy key mode temp 16251.000000 16251.000000 16251.0 16251.000000 16251.000000 16251.000000 16251.000000 16251.00000 count 16251.0 124.71646 0.0 241.830750 0.813450 0.0 5.313396 -9.928168 0.679404 232.870017 mean std 0.0 118.431612 1.993506 0.0 3.570588 5.008003 0.466720 116.066885 35.06637 0.000000 0.626000 0.626490 0.000000 0.000000 -47.403000 0.00000 min 0.0 0.0 25% 0.000000 2.000000 0.000000 171.668500 0.0 179.539140 0.0 -12.414000 98.49900 50% 0.0 225.488530 0.200000 0.0 5.000000 -8.904000 1.000000 216.317000 121.98800 -6.305000 145.66100 75% 0.0 279.104850 0.438000 0.0 9.000000 1.000000 269.012000 3029.080360 45.697000 11.000000 3.894000 1.000000 2999.583000 max 0.0 0.0 253.88700 # export to csv file In [24]: new df.to csv("successes55000 with spotify.csv") In []: