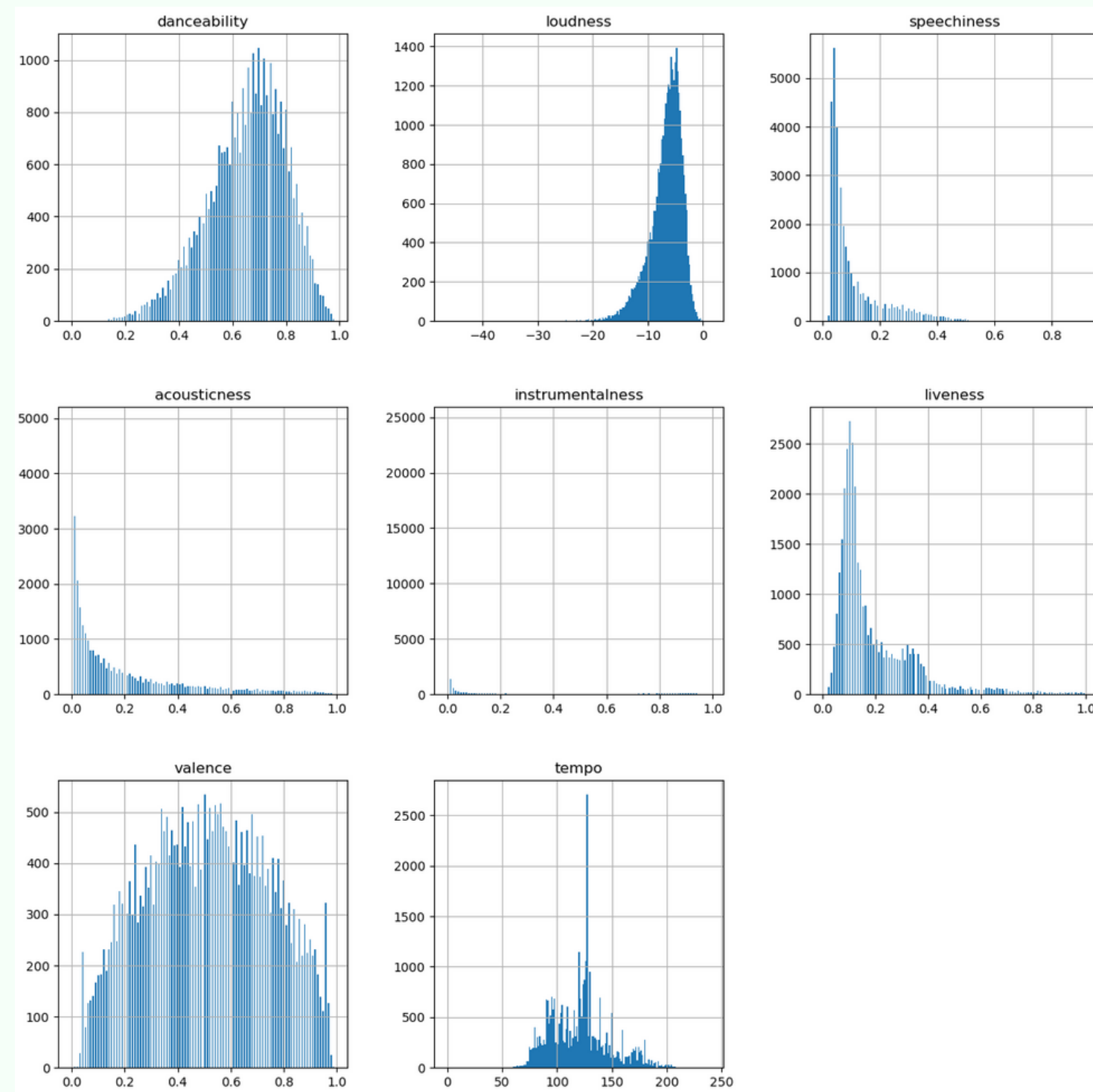


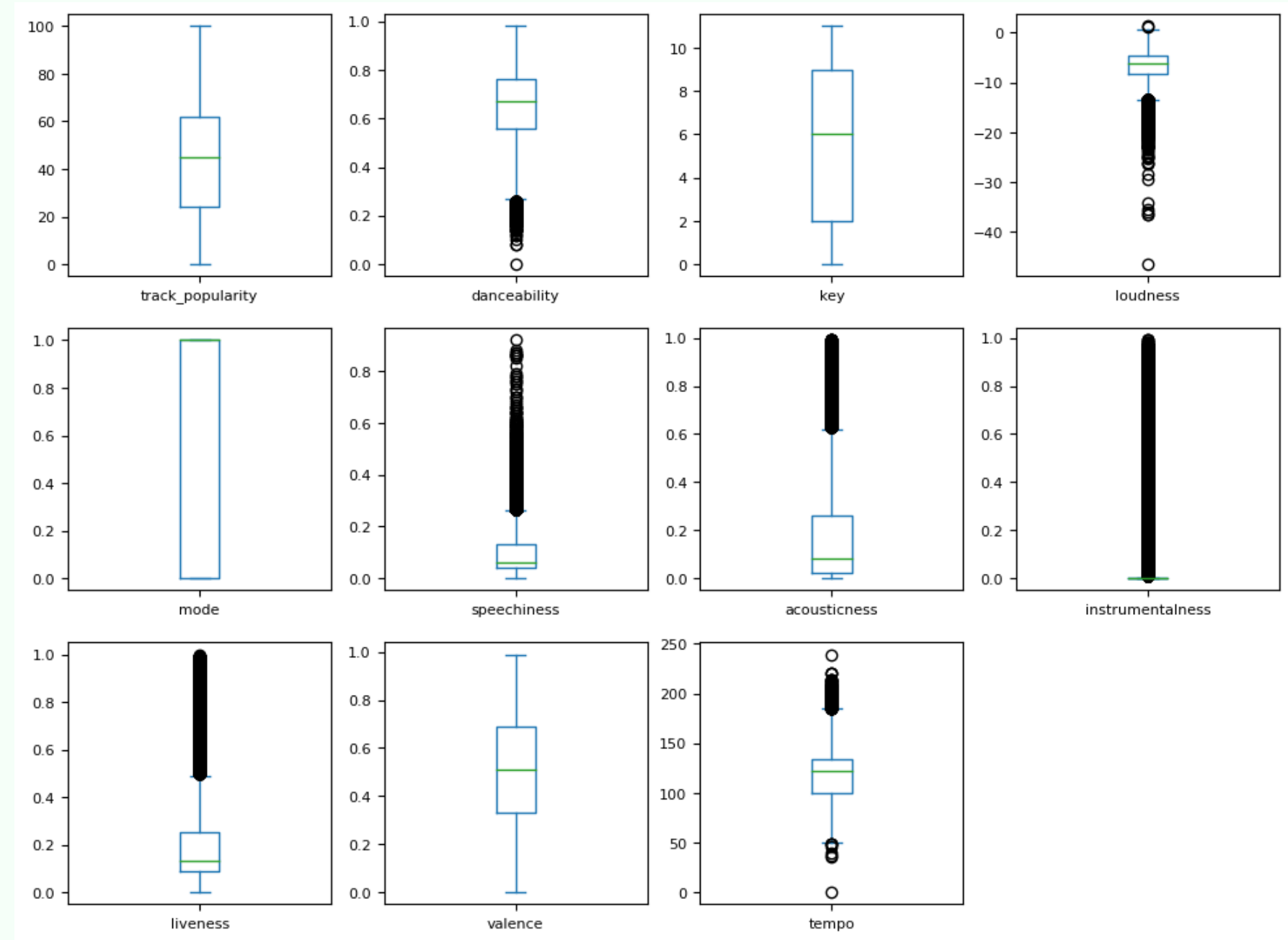
JIT PREDICTOR

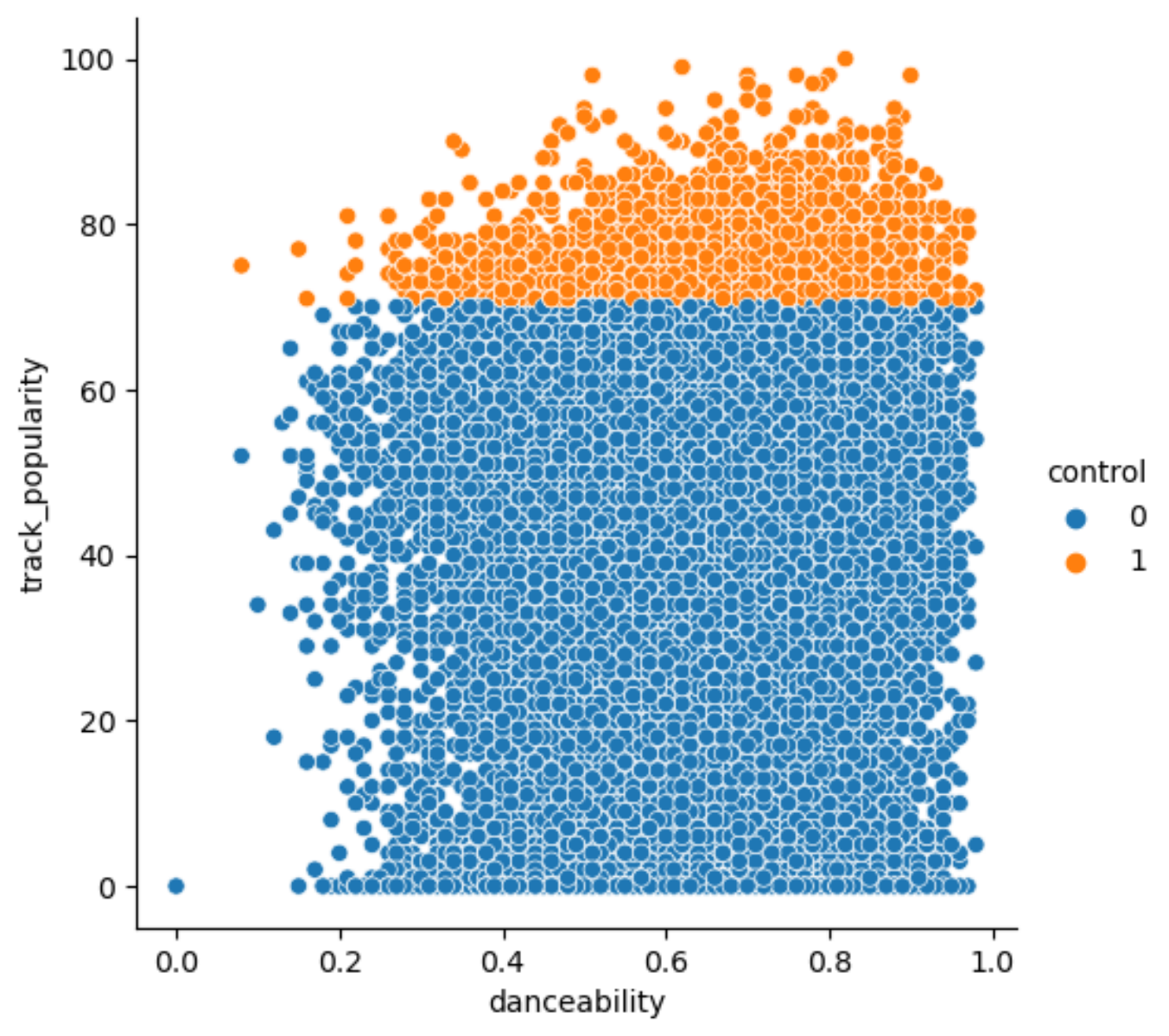
Binary classification on a Spotify Dataset

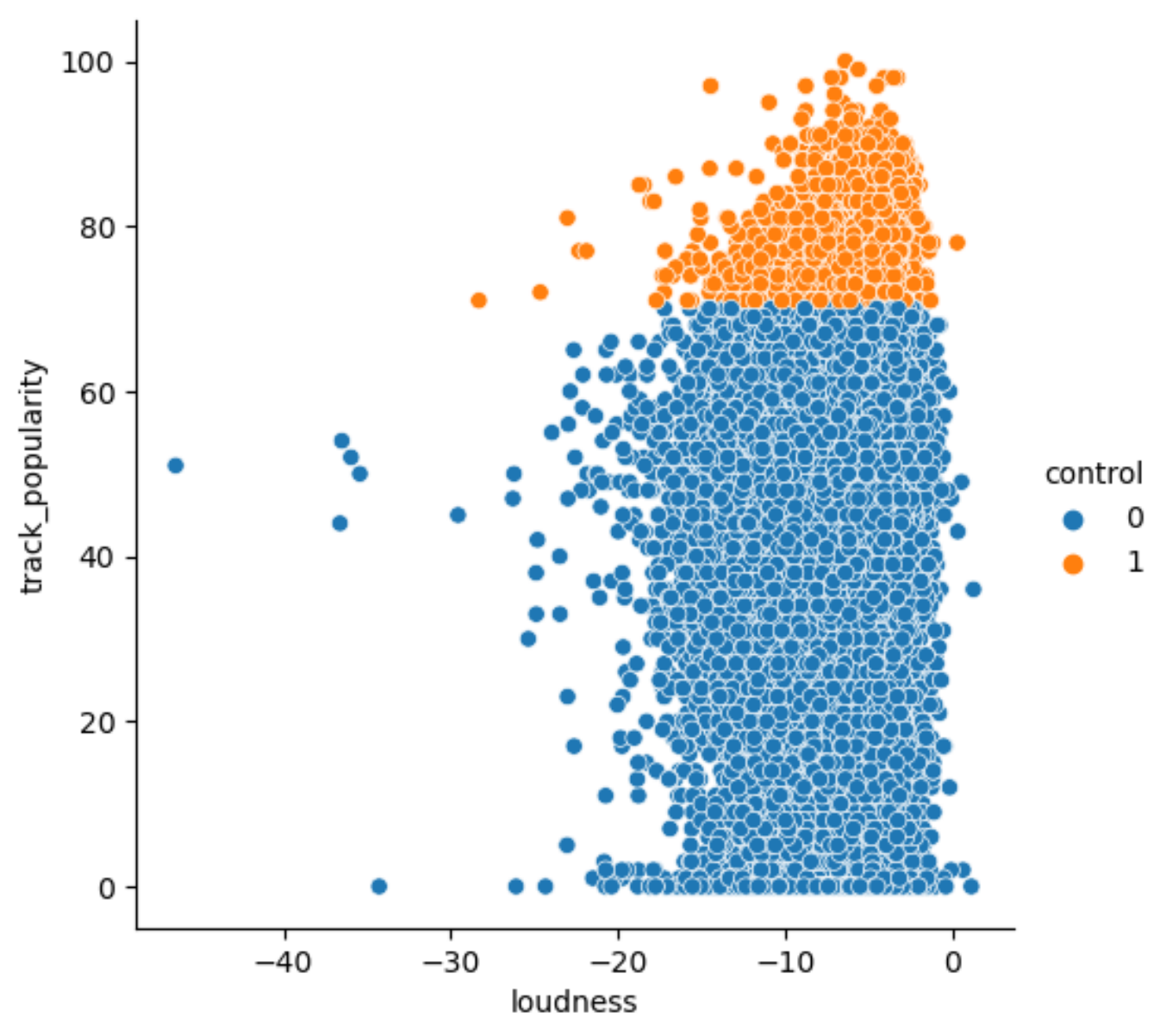
Texts

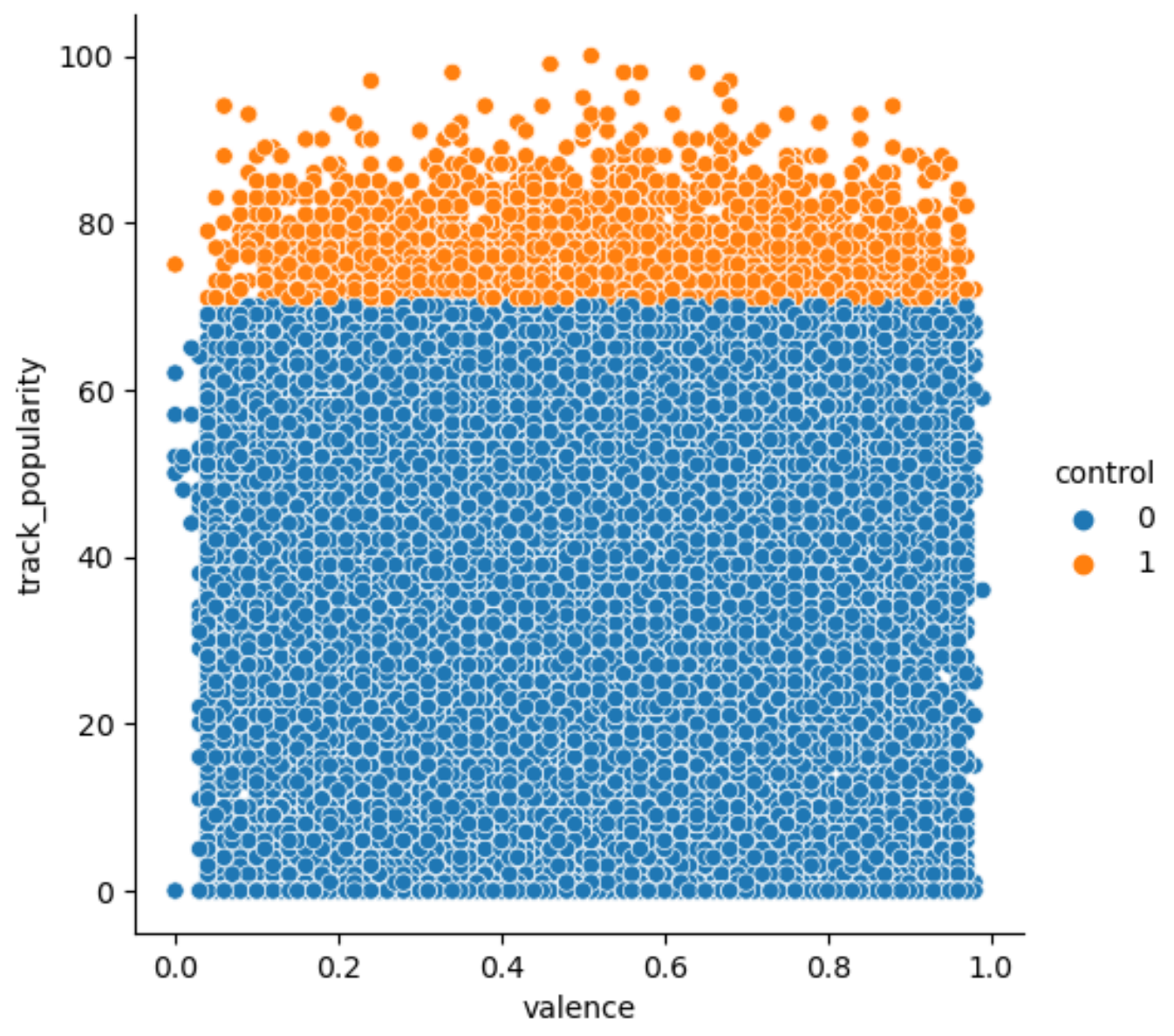
```
columns_numericas = df.select_dtypes(include=[ 'float64'])
```











No vemos grandes diferencias en cuanto a correlaciones x feature y grupo de control. Si acaso :

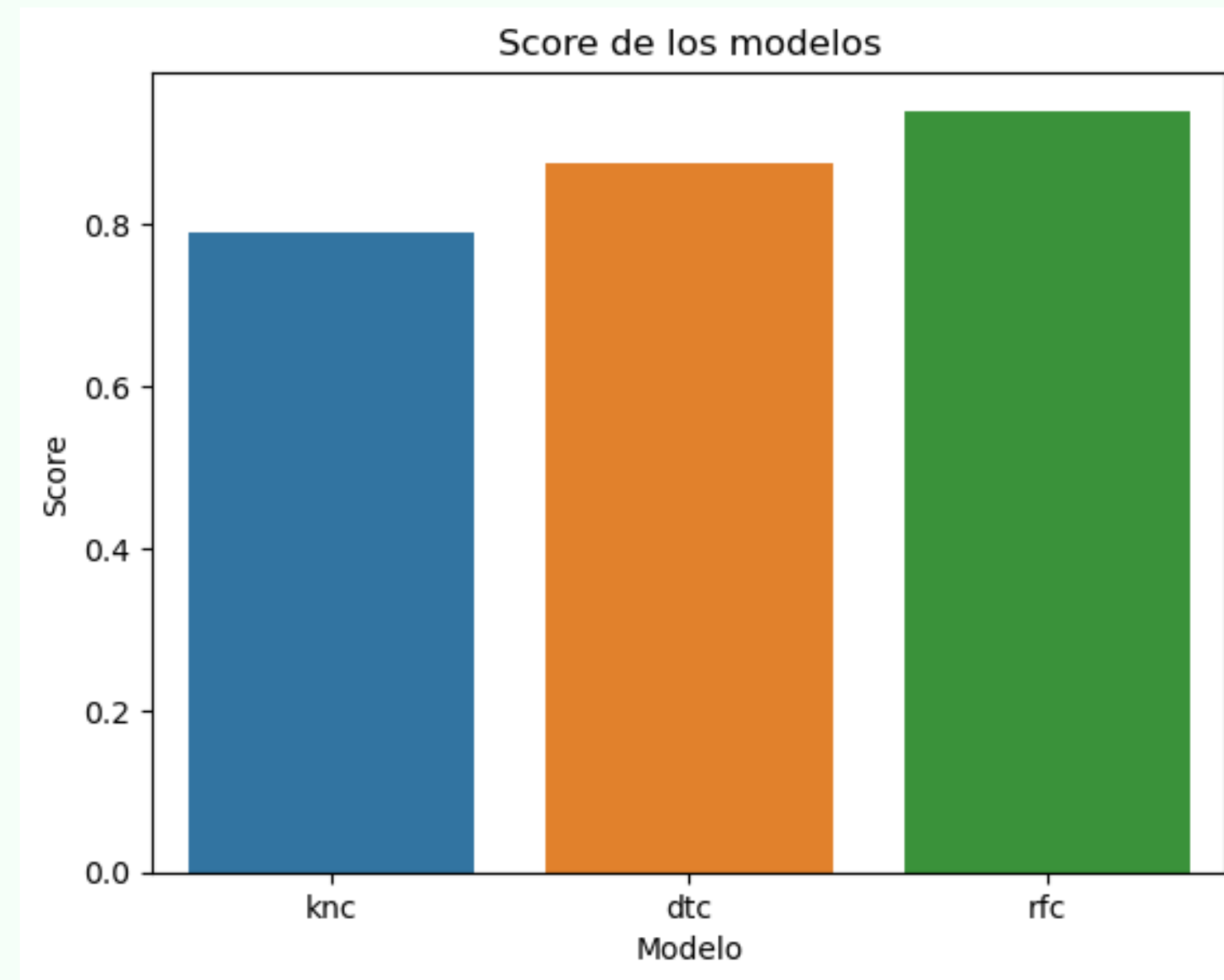
Danceability : 0.13 TOP vs 0.03 BOTTOM ...las canciones TOP suelen ser ligeramente más bailables.

Speechness : 0.07 TOP vs 0.004 BOTTOM ...las canciones TOP suelen incluir más partes "habladas", mas "Lyrics".

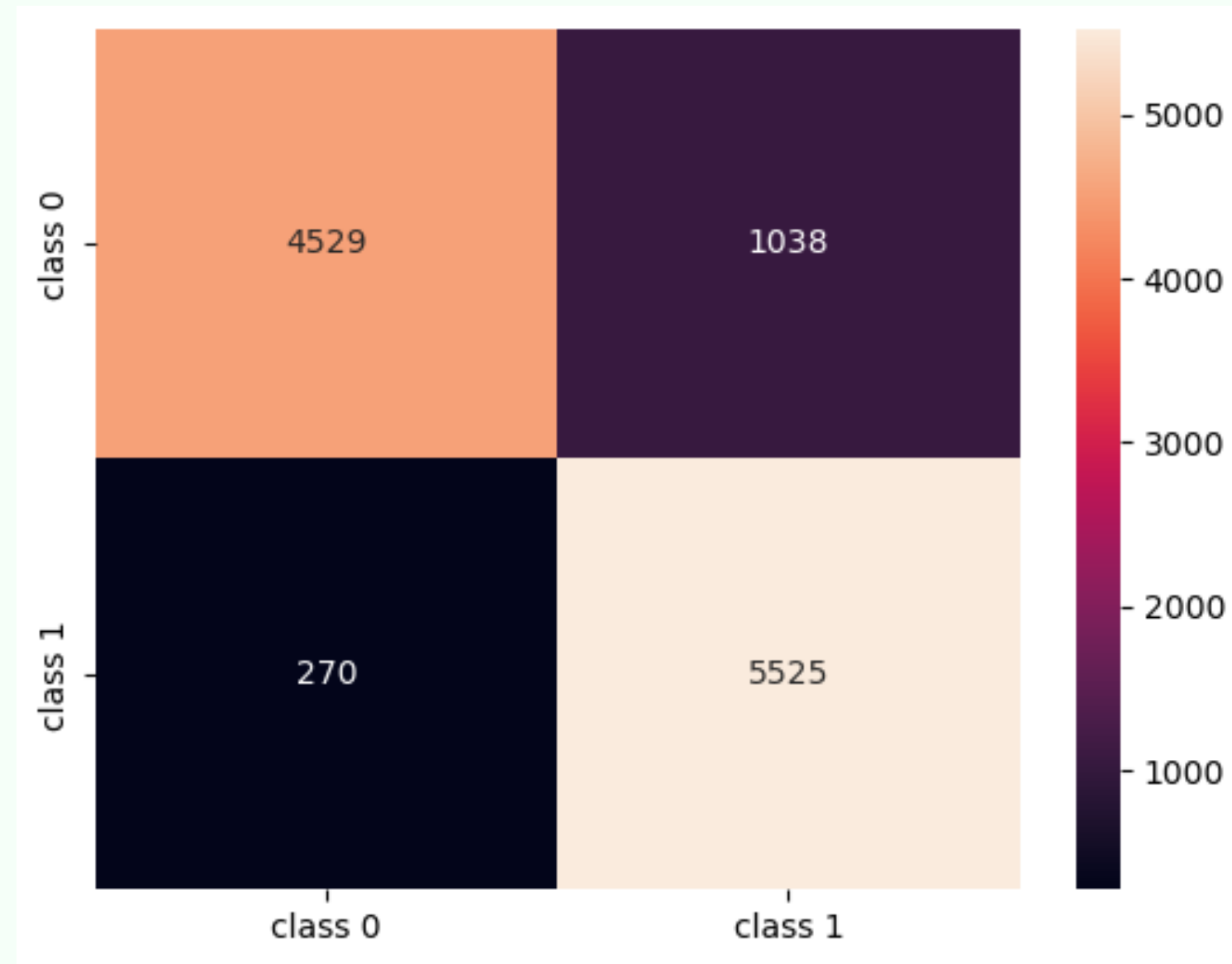
Instrumentalness : -0.014 TOP vs - 0.09 BOTTOM .. las canciones TOP son menos instrumentales que las canciones BOTTOM.
Confirma en cierta forma la importancia de las lyrics en cuanto a la popularidad de una canción (que no calidad)

Valence : -0.051 TOP vs 0.02 BOTTOM ...las canciones TOP guardan una mínima correlación negativa con Valence
y las canciones BOTTOM guardan una mínima correlación positiva con VALENCE.
las canciones TOP suelen ser más alegres que las canciones BOTTOM.

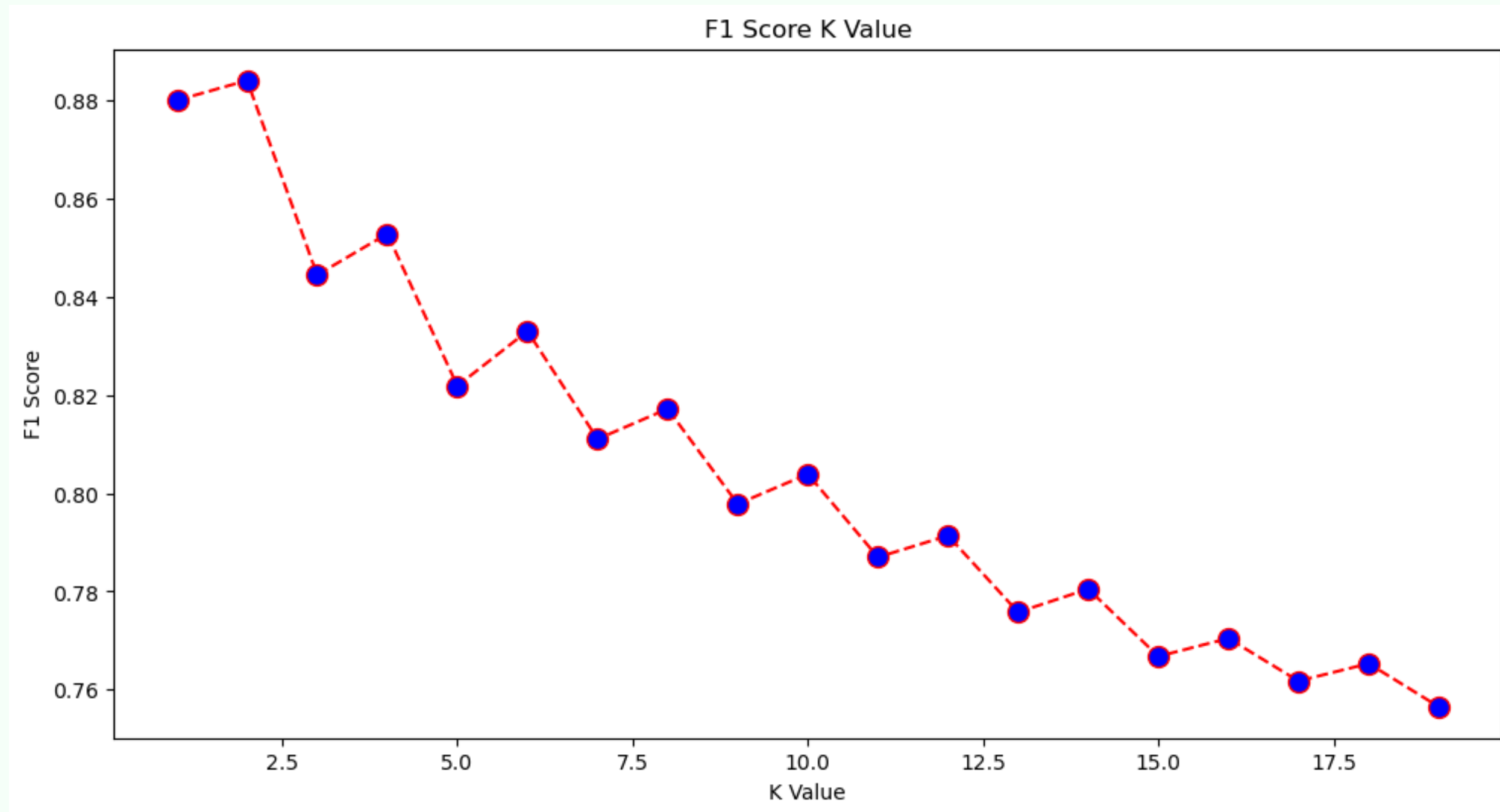
T



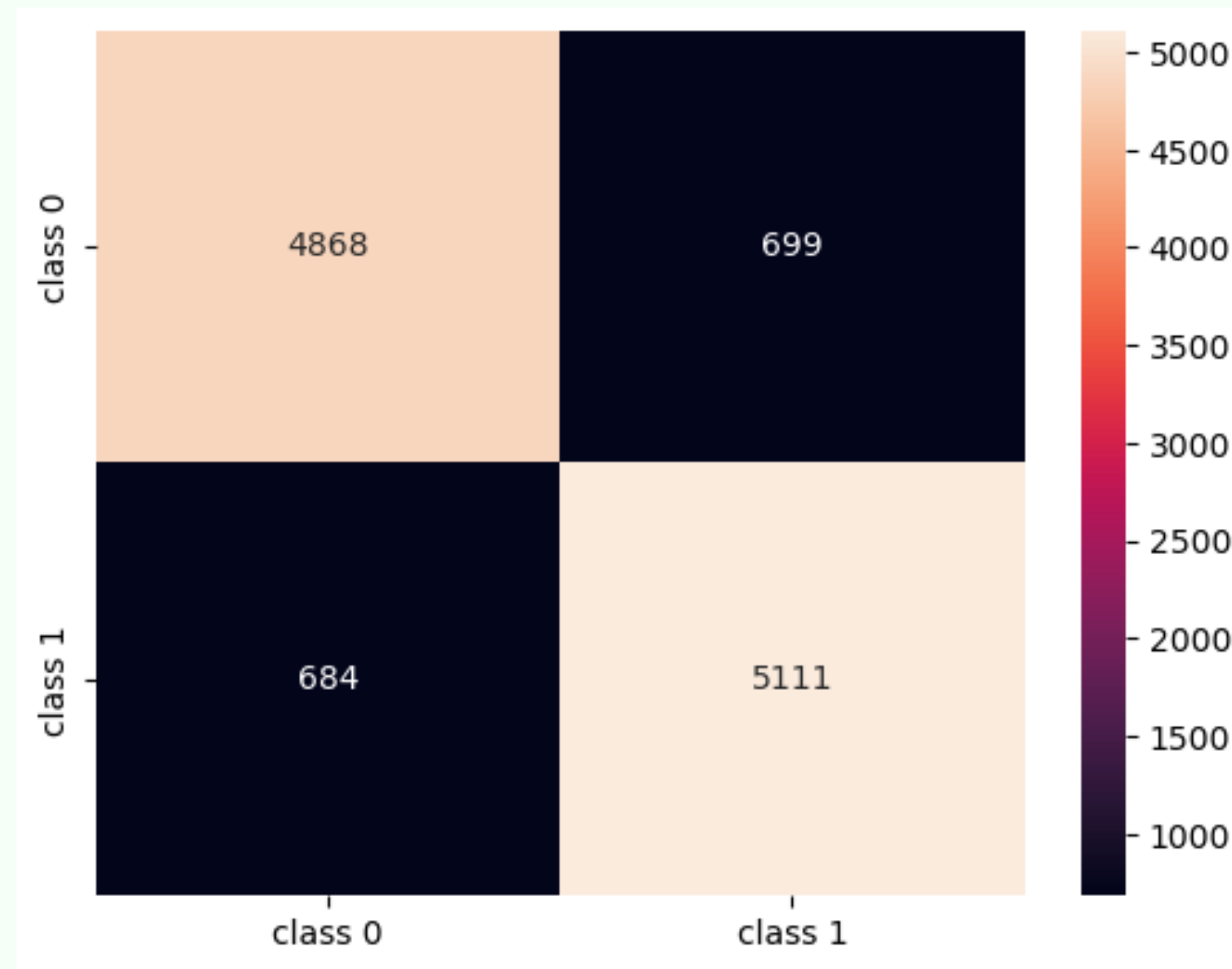
Scores análisis previo Modelos CrossValidation



Heatmap Accuracy para Best Model



Método K.Elbow para k.óptimo = 2



Heatmap Accuracy para K = 2

	precision	recall	f1-score	support
0	0.94	0.81	0.87	5567
1	0.84	0.95	0.89	5795
accuracy			0.88	11362
macro avg	0.89	0.88	0.88	11362
weighted avg	0.89	0.88	0.88	11362
	precision	recall	f1-score	support
0	0.88	0.87	0.88	5567
1	0.88	0.88	0.88	5795
accuracy			0.88	11362
macro avg	0.88	0.88	0.88	11362
weighted avg	0.88	0.88	0.88	11362

	popularity	danceability	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	control	predicted_popularity
0	68	0.48	4	-10.06	1	0.04	0.69	0.00	0.12	0.14	133.41	0	0
1	50	0.57	3	-10.29	1	0.03	0.48	0.00	0.10	0.52	140.18	0	0
2	57	0.41	3	-13.71	1	0.03	0.34	0.00	0.09	0.14	139.83	0	0
3	58	0.39	10	-9.85	1	0.04	0.81	0.00	0.08	0.51	204.96	0	0
4	54	0.43	6	-5.42	0	0.03	0.07	0.02	0.11	0.22	171.86	0	0

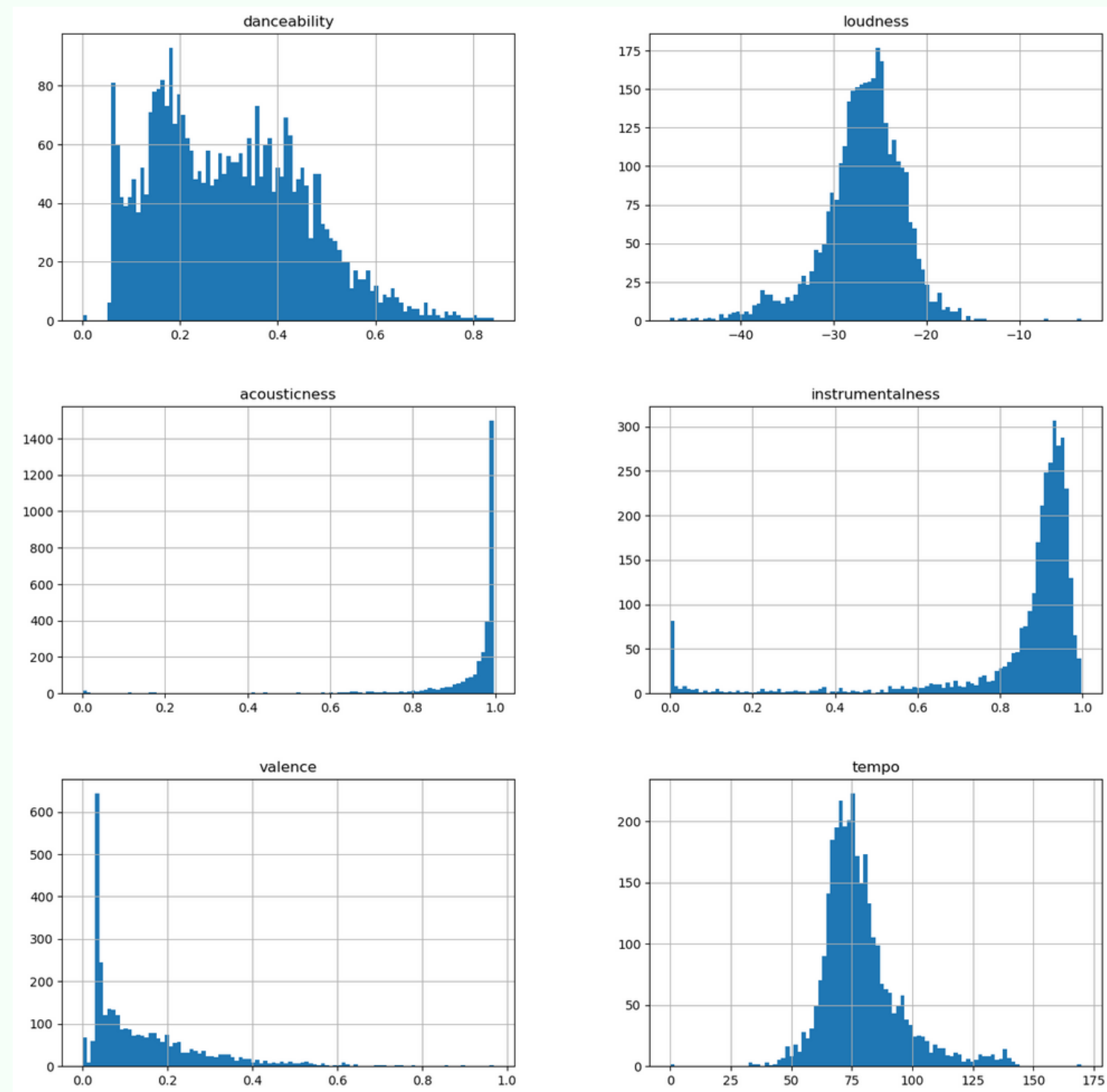
```
df2['control'] = df2['popularity'].apply(lambda x: 1 if x > 70 else 0)
```

```
predicted_popularity = best_model_knn.predict(df3)
```

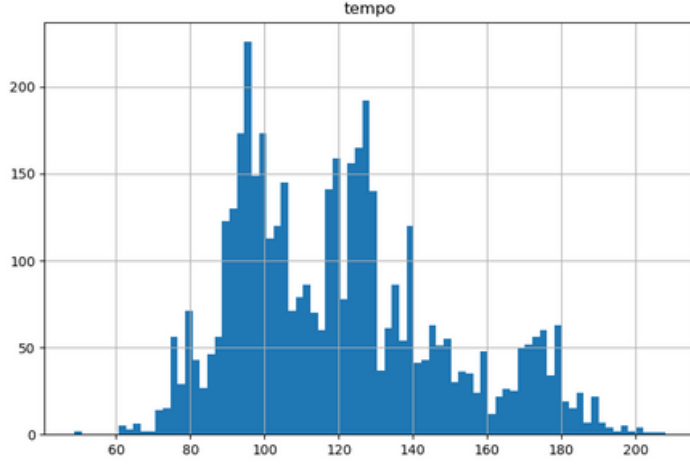
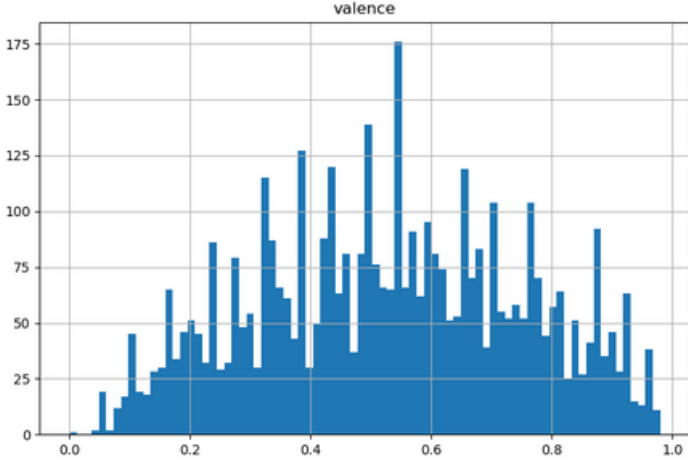
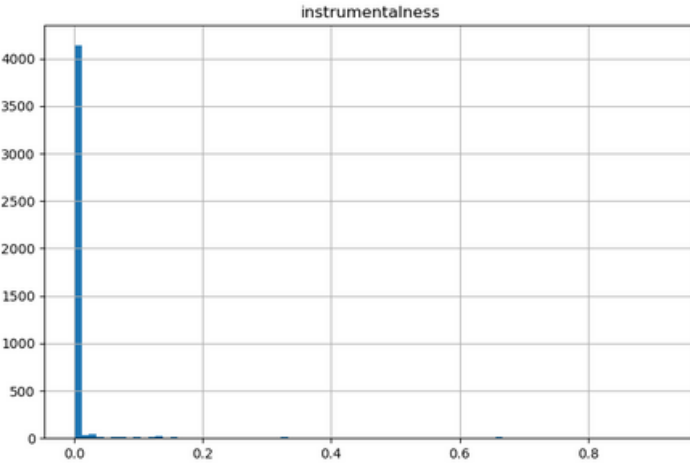
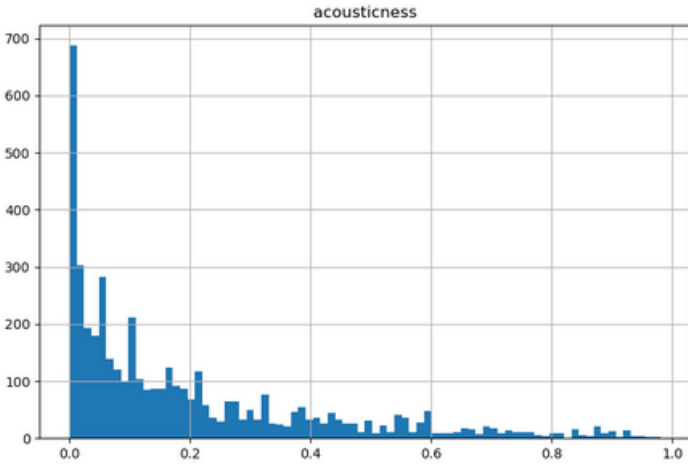
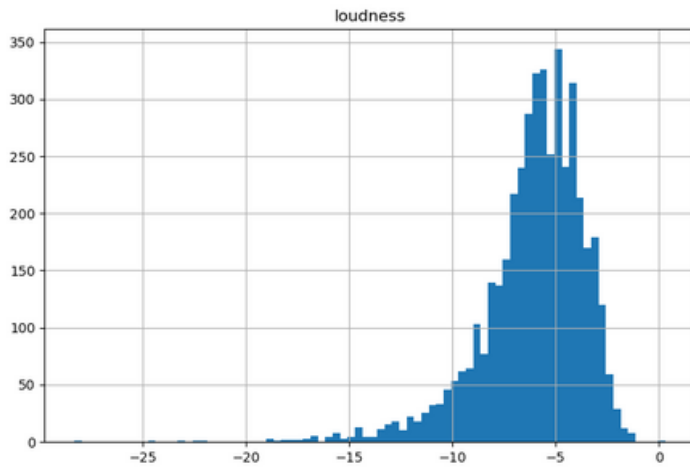
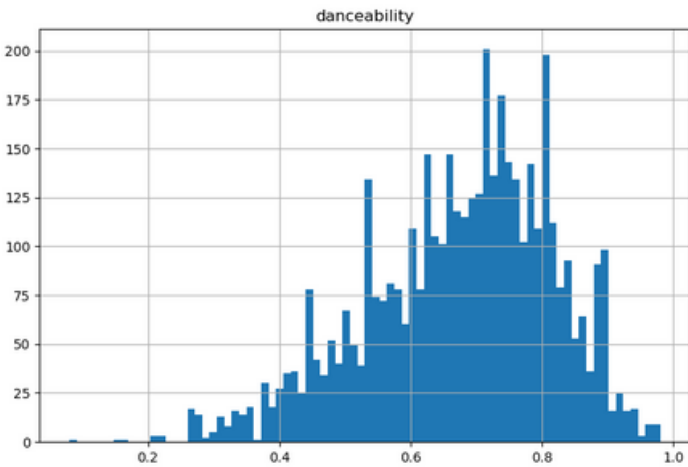
**# Buscamos las "perlas", canciones que por características (segun algoritmo de class por features
sónicas) deberían ser populares (predicted_popularity = 1) y que sin embargo no alcanzaron
popularidad (control = 0)**

df_pearl = df2.query('control == 0 & predicted_popularity == 1 & 30 <popularity < 60')

1	df_pear1.head()												
	popularity	danceability	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	control	predicted_popularity
2733	53	0.07	9	-24.01	0	0.04	0.86	0.92	0.10	0.05	80.49	0	1
2745	50	0.28	4	-14.32	0	0.03	0.66	0.98	0.10	0.08	39.37	0	1
2751	45	0.51	1	-23.25	1	0.04	0.99	0.88	0.14	0.05	73.36	0	1
2754	39	0.08	11	-21.73	1	0.04	0.85	0.83	0.10	0.03	56.58	0	1
2764	38	0.11	11	-22.85	1	0.04	0.92	0.89	0.09	0.03	72.66	0	1



HITS POPULARES



PERLAS
OLVIDADAS