Spotify Tracks DB

Data visualization

lets meet the data set

The dataset contains information about tracks (music) on Spotify

Accurate to the year 2021
Tabular data set

lets meet the data set

The dataset has 232,725 tracks (rows)
Each track has 18 feature (columns)

dtypes:
float64(9)
int64(2)
object(7)
memory usage: 32.0+ MB

No Null values

first 5 tracks:

5 rows × 18 columns pd.DataFrame ↗ CSV

	genre	artist_name	track_name	track_id	popularity	acousticness	danceabilit
0	Movie	Henri Salvador	C'est beau de faire un Show	0BRjO6ga9RKCKjfDqeFgWV	0	0.611	0
1	Movie	Martin & les fées	Perdu d'avance (par Gad Elmaleh)	0BjC1NfoEOOusryehmNudP	1	0.246	0
2	Movie	Joseph Williams	Don't Let Me Be Lonely Tonight	0CoSDzoNIKCRs124s9uTVy	3	0.952	0
3	Movie	Henri Salvador	Dis-moi Monsieur Gordon Cooper	0Gc6TVm52BwZD07Ki6tIvf	0	0.703	0
4	Movie	Fabien Nataf	Ouverture	0IuslXpMROHdEPvSl1fTQK	4	0.950	0

#	Column	Non-Null Count	Dtype
0	**genre**	232725 non-null	object
1	artist_name	232725 non-null	object
2	track_name	232724 non-null	object
3	track_id	232725 non-null	object
4	popularity	232725 non-null	int64
5	acousticness	232725 non-null	float64
6	danceability	232725 non-null	float64
7	duration_ms	232725 non-null	int64
8	energy	232725 non-null	float64
9	instrumentalness	232725 non-null	float64
10	key	232725 non-null	object
11	liveness	232725 non-null	float64
12	loudness	232725 non-null	float64
13	mode	232725 non-null	object
14	speechiness	232725 non-null	float64
15	tempo	232725 non-null	float64
16	time_signature	232725 non-null	object
17	valence	232725 non-null	float64

Values destrebution

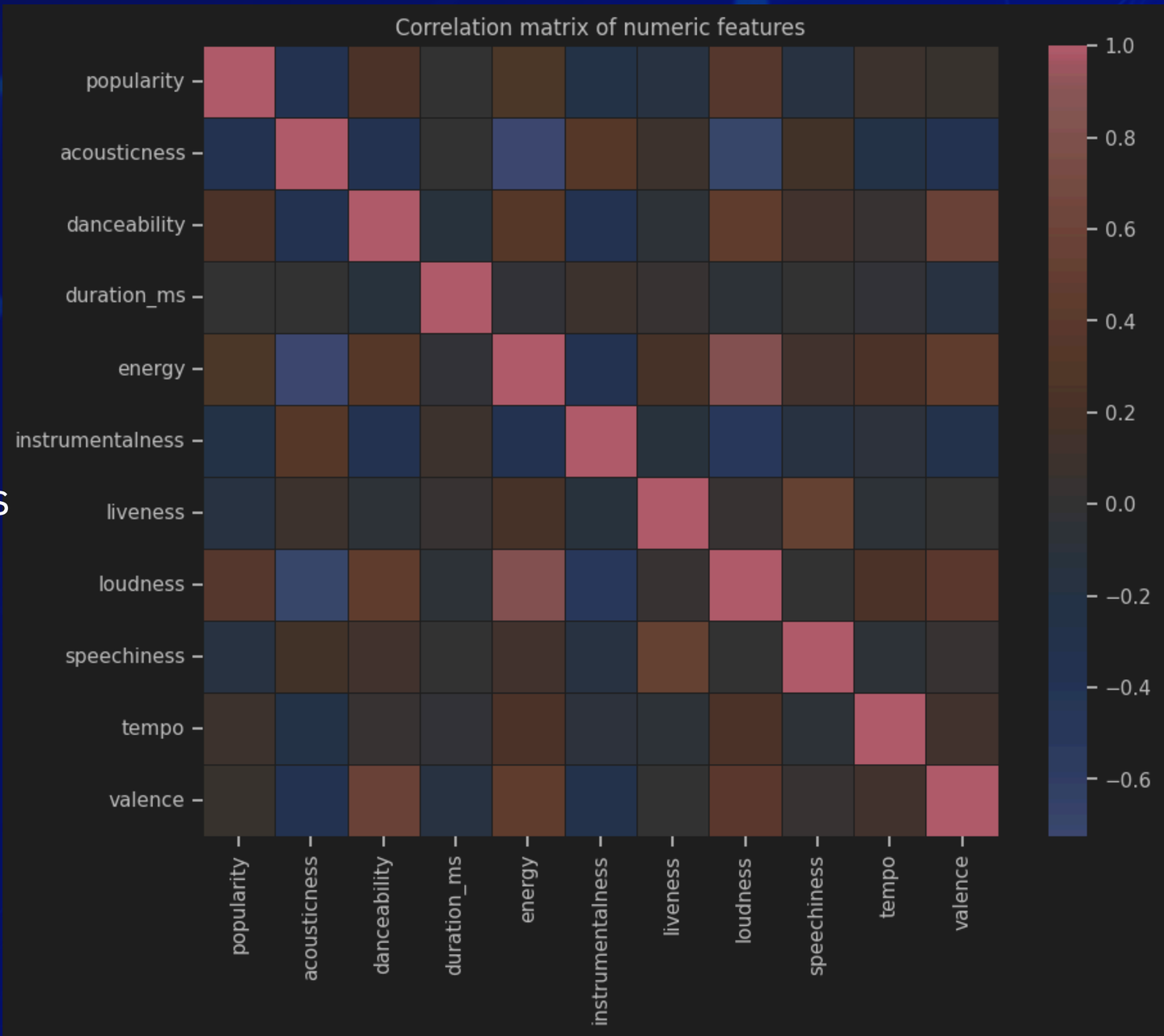
The feature distribution is not normalized.
The scales of the values are different (negative, large, small and ranges)
For some, the mean is not close to the median (good if some geners are far)

11 rows | 11 rows × 8 columns pd.DataFrame | CSV

	count	mean	std	min	25%	50%	75%	max
popularity	232725.0	41.127502	18.189948	0.00000	29.0000	43.000000	55.0000	100.000
acousticness	232725.0	0.368560	0.354768	0.00000	0.0376	0.232000	0.7220	0.996
danceability	232725.0	0.554364	0.185608	0.05690	0.4350	0.571000	0.6920	0.989
duration_ms	232725.0	235122.339306	118935.909299	15387.00000	182857.0000	220427.000000	265768.0000	5552917.000
energy	232725.0	0.570958	0.263456	0.00002	0.3850	0.605000	0.7870	0.999
instrumentalness	232725.0	0.148301	0.302768	0.00000	0.0000	0.000044	0.0358	0.999
liveness	232725.0	0.215009	0.198273	0.00967	0.0974	0.128000	0.2640	1.000
loudness	232725.0	-9.569885	5.998204	-52.45700	-11.7710	-7.762000	-5.5010	3.744
speechiness	232725.0	0.120765	0.185518	0.02220	0.0367	0.050100	0.1050	0.967
tempo	232725.0	117.666585	30.898907	30.37900	92.9590	115.778000	139.0540	242.903
valence	232725.0	0.454917	0.260065	0.00000	0.2370	0.444000	0.6600	1.000

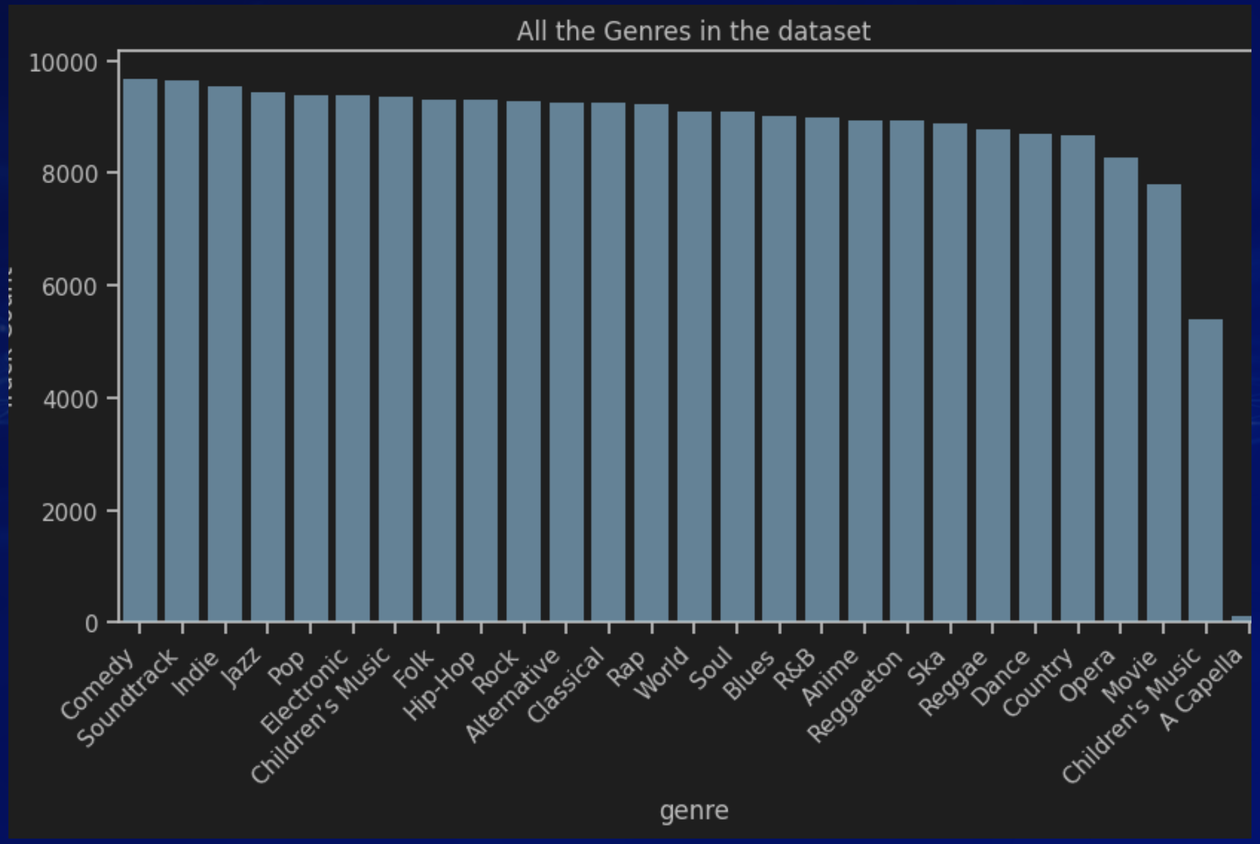
Feature correlation

From my perspective, it's safe to say the correlation between our features is not significant.
Which means that no feature is unnecessary because it can be represented by another.



Lets talk genres!

We got 27 genres
Each has up to 10,000 tracks
It contains very few aCapella tracks



I exported a CSV that contains the mean, median, and std of each value in each genre

It seems like an interesting thing to plot

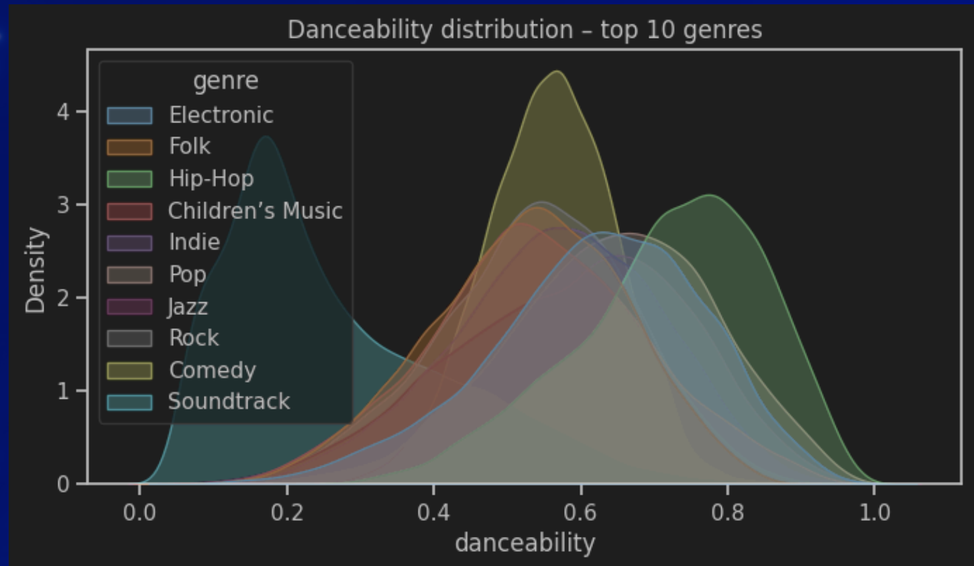
1-11 27 rows × 44 columns pd.DataFrame

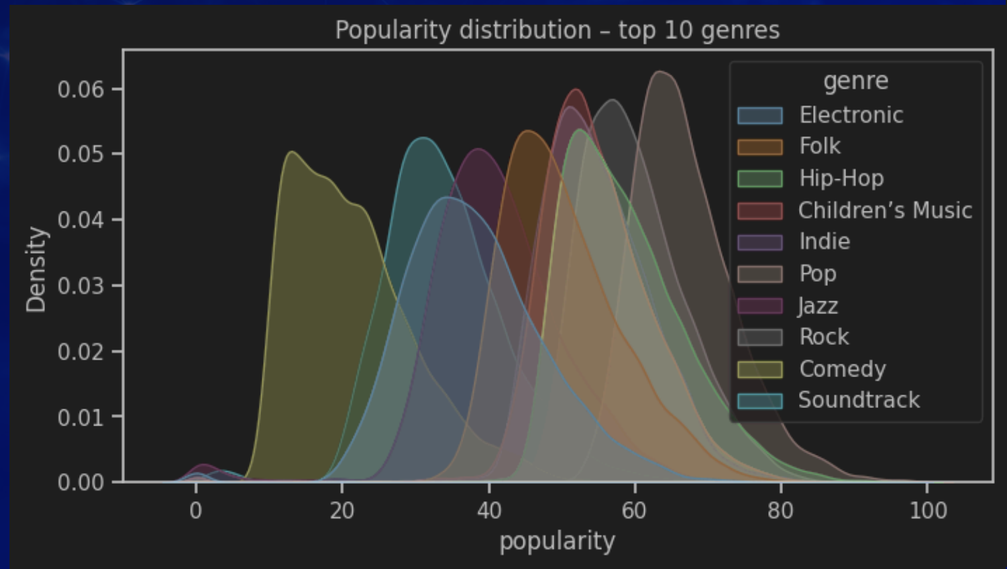
genre	popularity count	mean	median	std	acousticness count	mea
Comedy	9681	21.34	20.0	8.43	9681	
Soundtrack	9646	33.95	33.0	8.64	9646	
Indie	9543	54.70	54.0	7.36	9543	
Jazz	9441	40.82	40.0	9.59	9441	
Pop	9386	66.59	66.0	7.25	9386	
Electronic	9377	38.06	37.0	9.74	9377	
Children's Music	9353	54.66	54.0	7.85	9353	
Folk	9299	49.94	49.0	8.22	9299	
Hip-Hop	9295	58.42	57.0	8.27	9295	
Rock	9272	59.62	59.0	7.47	9272	
Alternative	9263	50.21	49.0	7.66	9263	

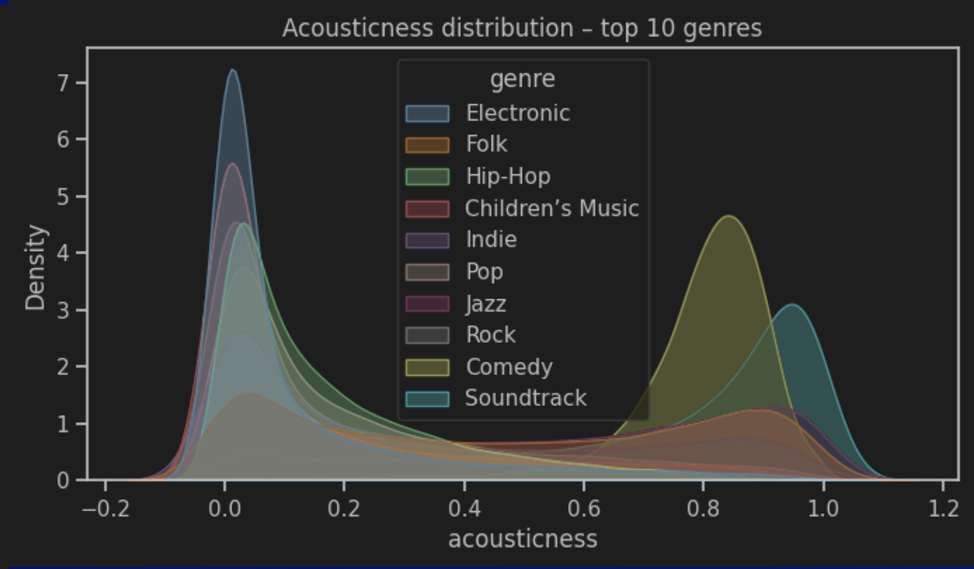
Distributions group by genre

Distribution of 10 genres feature values plotted
Only 10 because it makes a big mess to plot them all

hypothesis: those will be the most significant features because it seems like every genre has its range



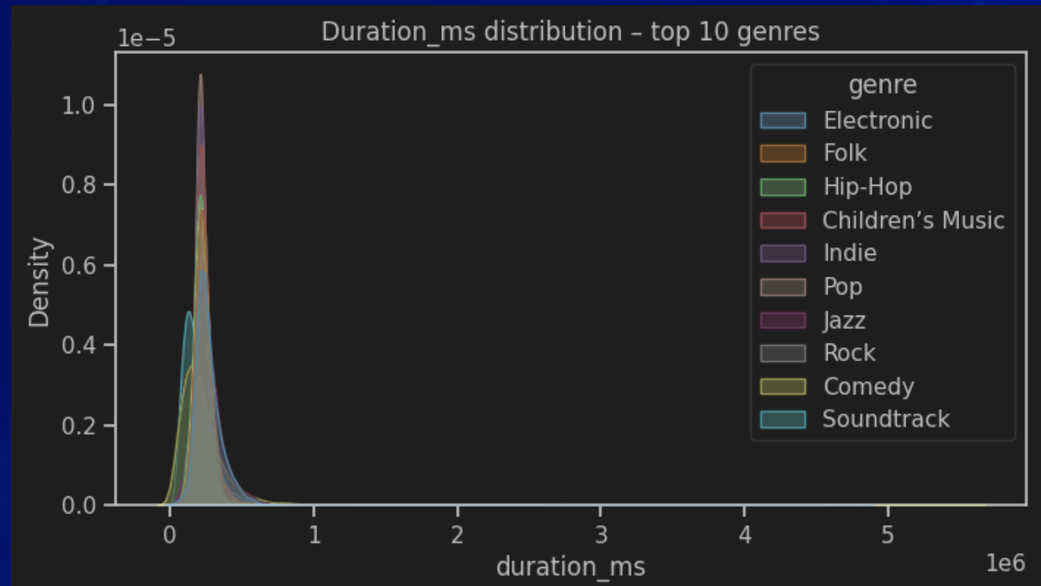


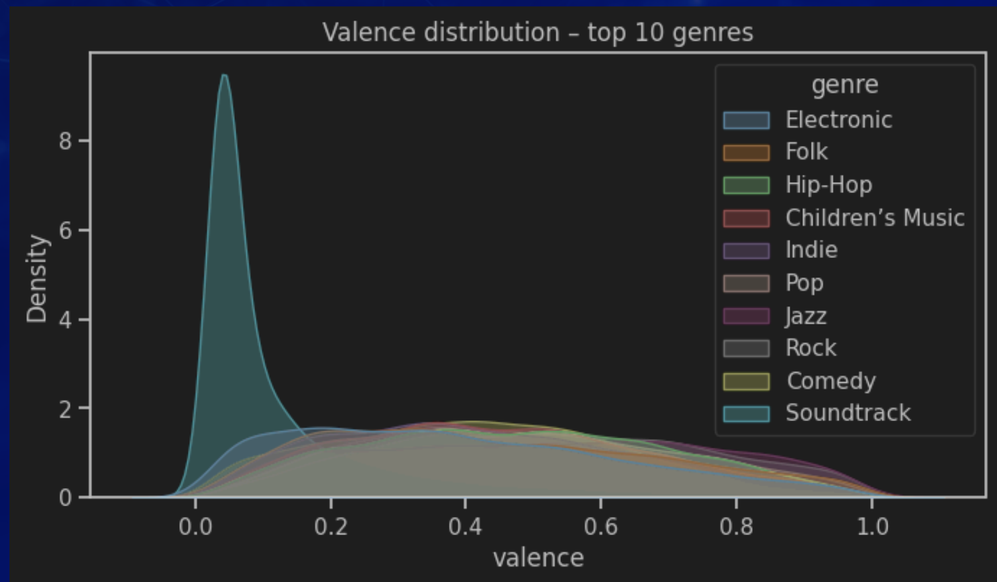


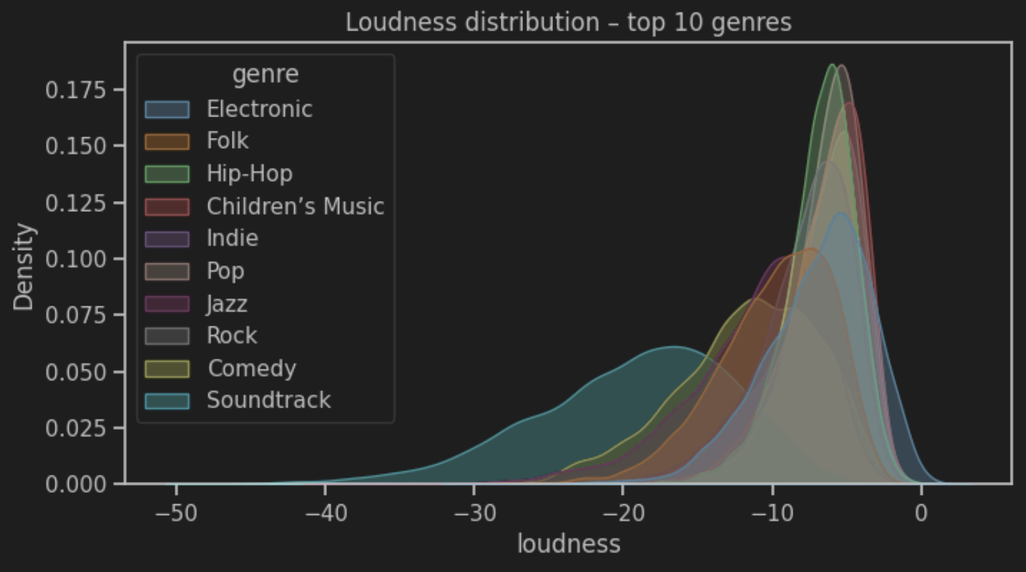
Lets talk genres!

Distribution of 10 genres feature values plotted
Only 10 because it makes a big mess to plot them all

Hypothesis: those will be the most insignificant features because it seems all the genre ranges are the same







I plotted all the distributions
it's all in the notebook

Just 4 Fun

Just because I was curious I wanted to see
The most popular artists in the dataset

It provides more information about
the data, but it's irrelevant

This information got me thinking about how
important the artist name to the genre classification

Top 15 consistently popular artists (≥25 tracks), JUST 4 FUN:

artist_name	mean_popularity	n_tracks
Post Malone	76.0	85
Juice WRLD	75.5	40
Offset	74.7	58
Khalid	74.4	32
XXXTENTACION	73.0	106
Metro Boomin	73.0	41
6ix9ine	72.9	26
Ed Sheeran	72.9	47
Ariana Grande	72.6	142
Hailee Steinfeld	72.3	25
Bad Bunny	72.0	34
Lil Peep	71.7	56
Alec Benjamin	71.7	25
Camila Cabello	71.0	50
Marshmello	70.8	34

Feature Engineering

Milliseconds to minutes:

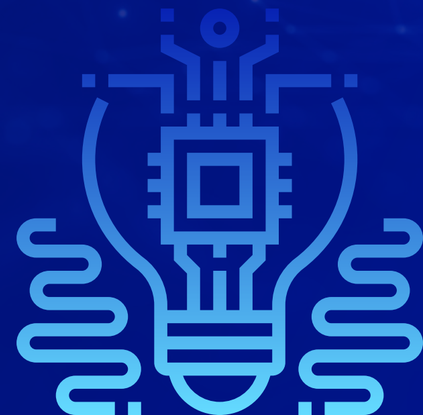
Milliseconds are not intuitive to the user.

The huge numbers will cause a longer training time.

Acoustic-instrumental synergy:

acousticness * instrumentalness

Might be useful for separating classic style from modern



Feature Engineering

Object features into numeric features:

Mode - binary 0 or 1

Key - mapping 1-12, then apply cos and sin encoding

Time signatures - numerator*2 / denominator

TF-IDF:

Artist name - apply mean TF-IDF

Track name - apply mean TF-IDF

Feature Engineering

Sumery:
18 features in total all of them are 1D and numeric.

"duration_min",
"danceability", "energy", "liveness", "speechiness",
"valence", "tempo", "acousticness",
"instrumentalness", "loudness", "popularity",
"ts_value", "is_major", "key_sin" ,"key_cos" ,
"acoustic_instrumentalness",
"track_tfidf_mean", "artist_tfidf_mean",

AdaBoost.
Random-Forest.
LightGBM.

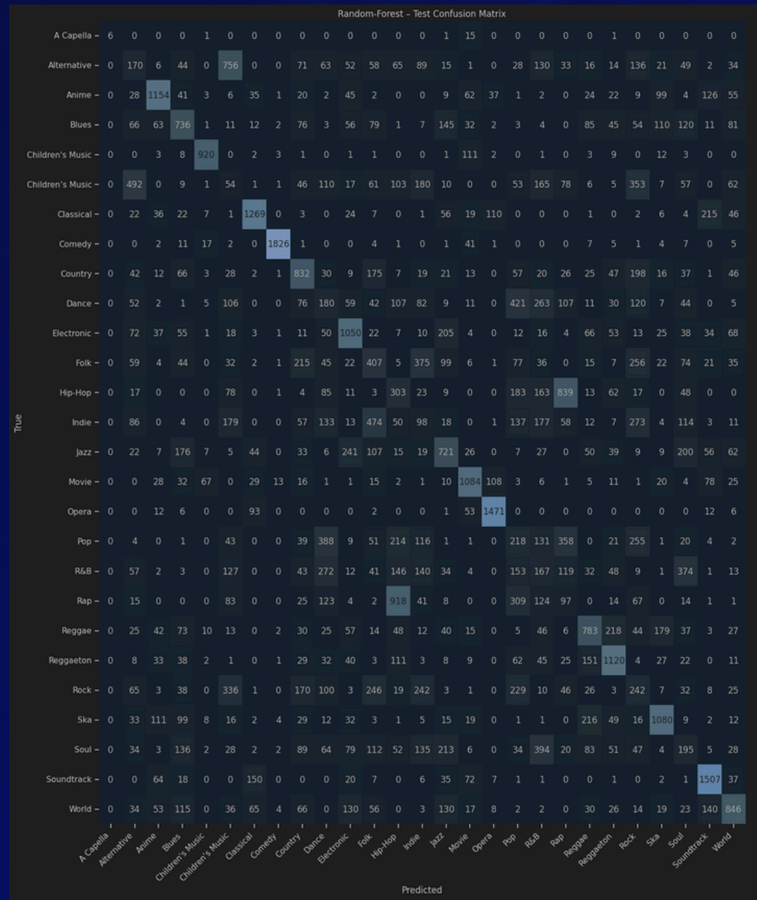
	acc	F1
LightGBM	42.5%	42.7%
Random-Forest	41.0%	41.5%
AdaBoost	20.4%	19.0%

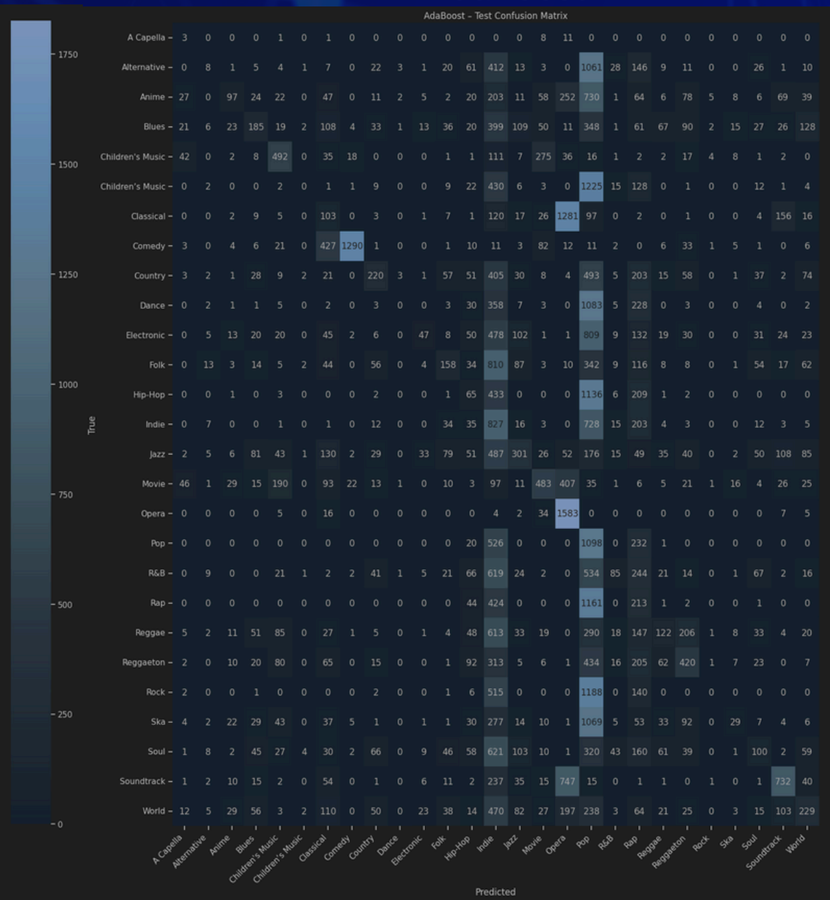
Random-Forest

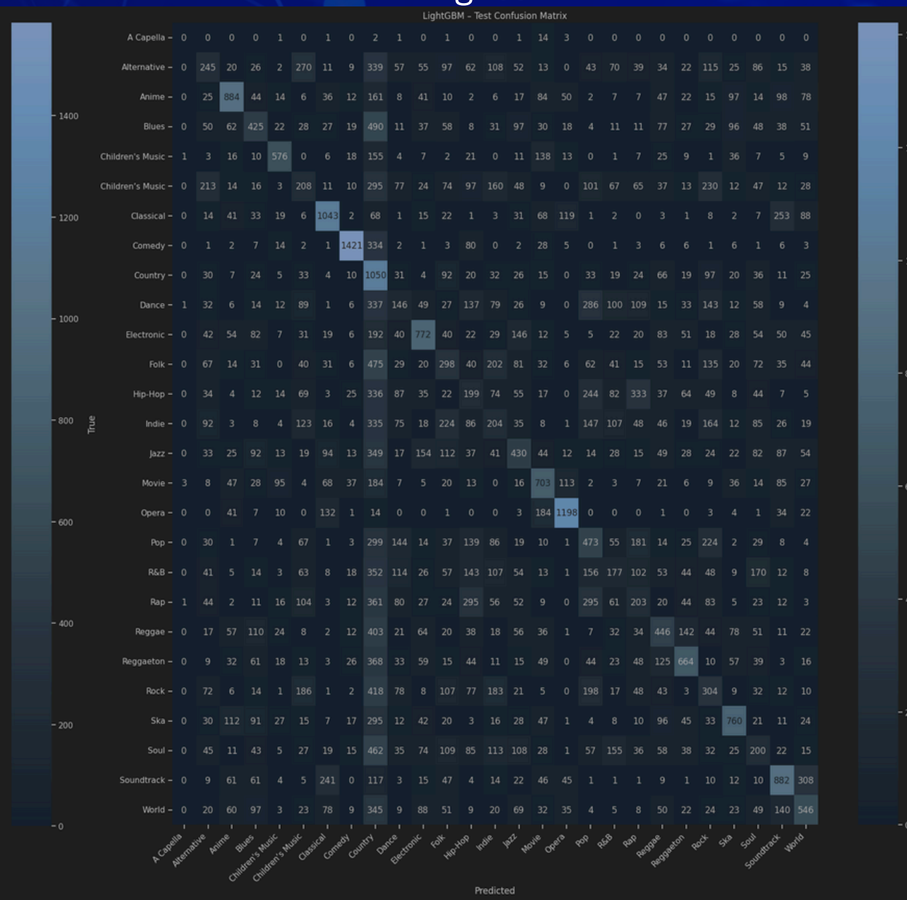
***pop + indie

AdaBoost

LightGBM





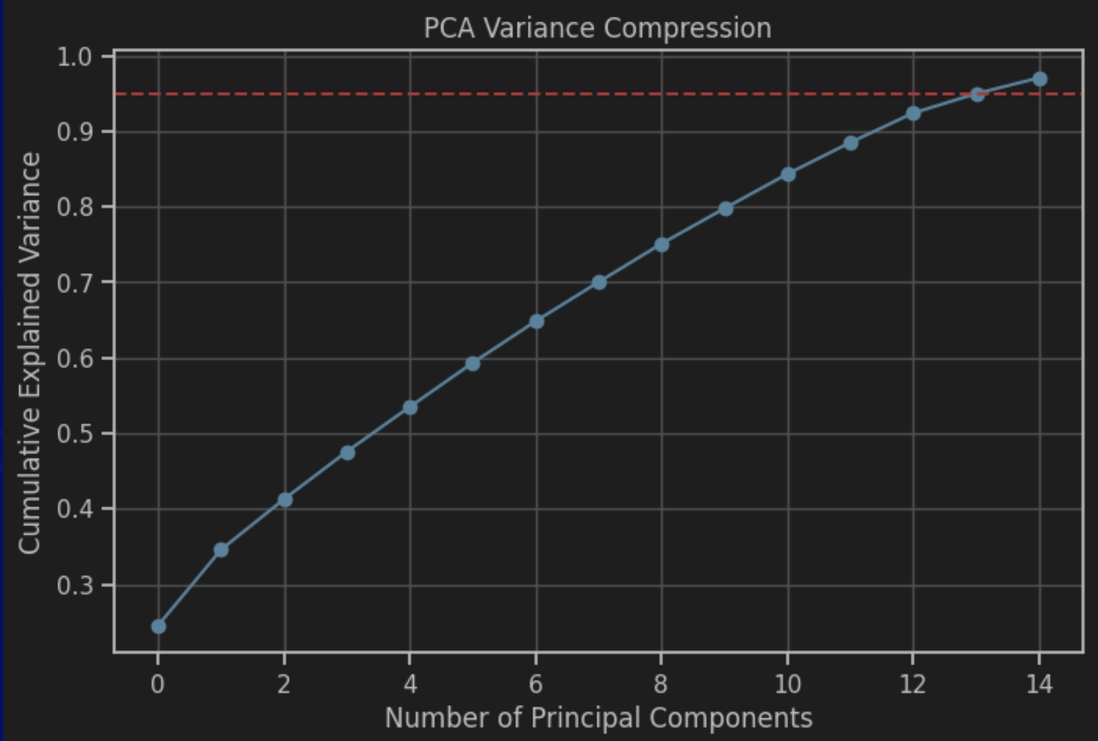


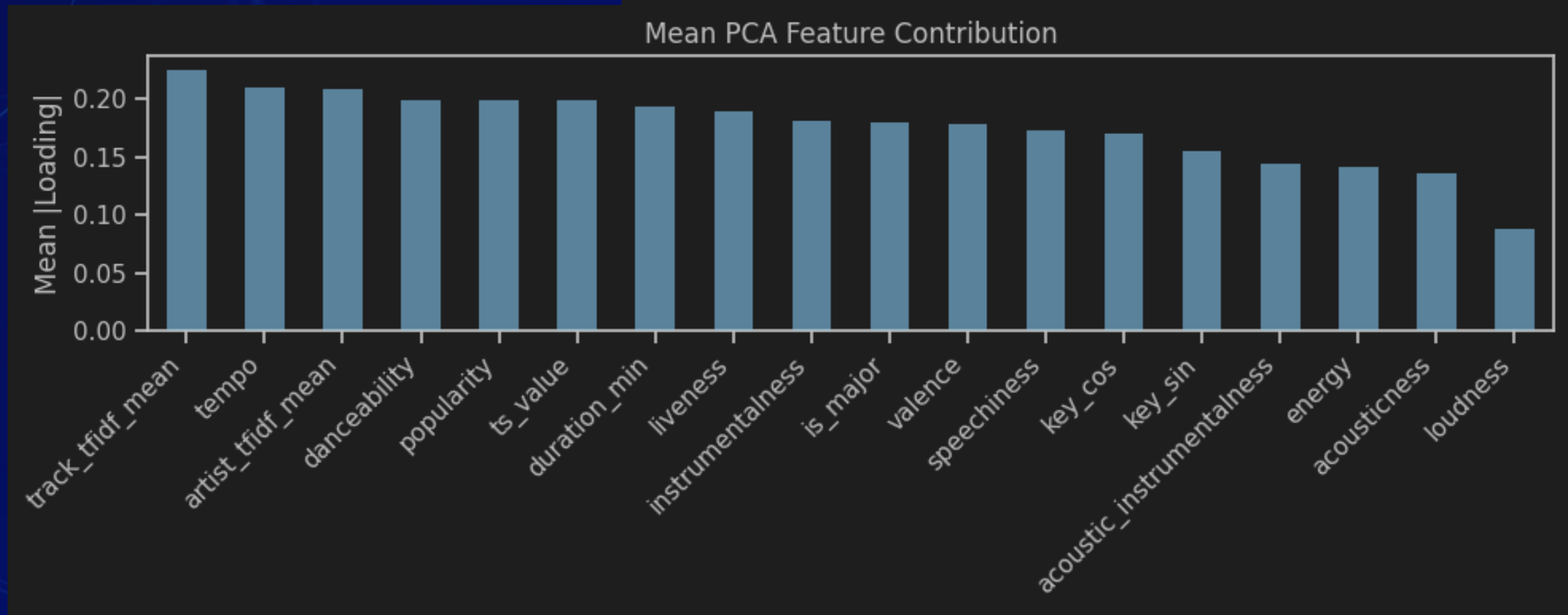
PCA:

Original numeric features: 18
PCA kept components: 15

My hypothesis:
Strong features-
- popularity
- danceability
- acousticness

Weak features-
- velence
- loudness

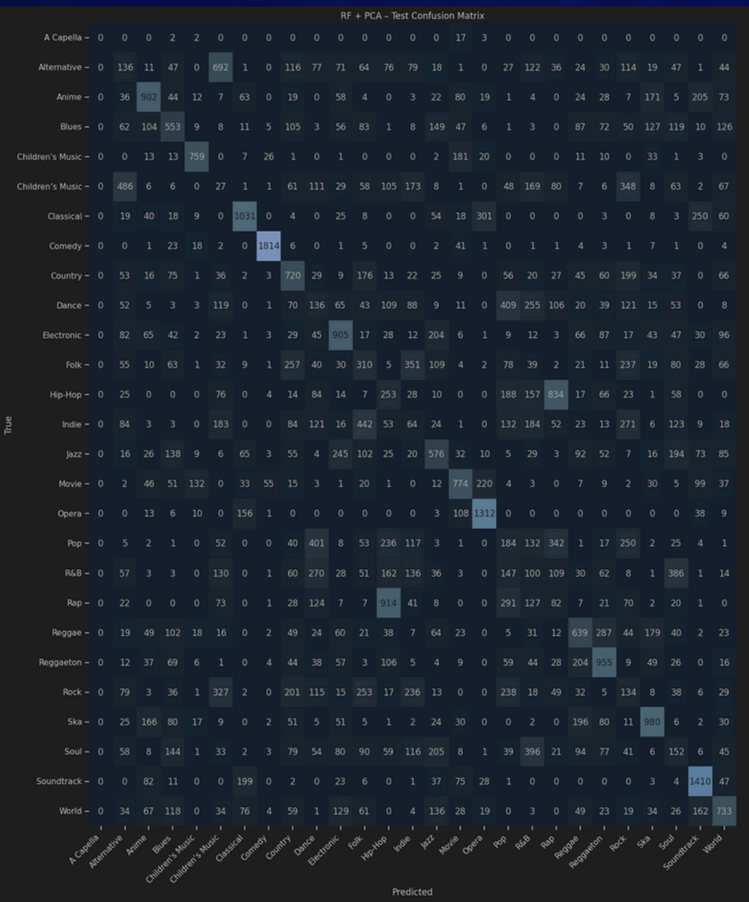




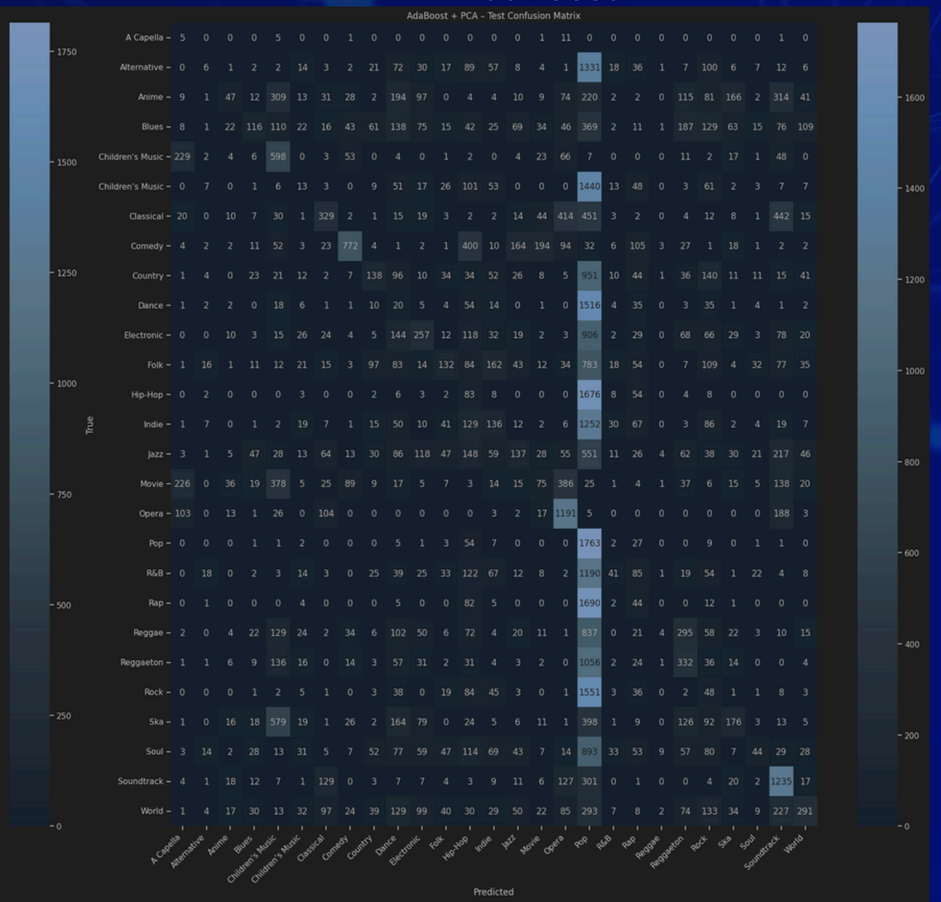
PCA:
AdaBoost.
Random-Forest.

	acc	F1
Random Forest + PCA	34.0%	33.7%
AdaBoost + PCA	17.3%	14.5%

Random-Forest



AdaBoost



I wasn't surprised that the performance is lower.

But I didn't expect such a drop

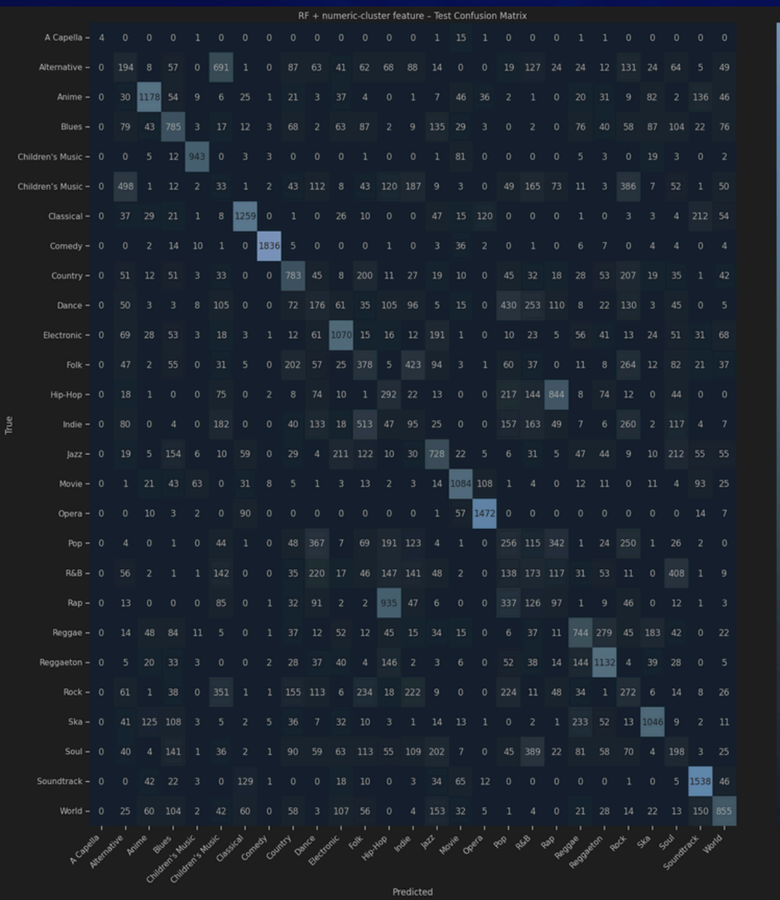
K-mean:
AdaBoost.
Random-Forest.

Run the data through K-means algorithm with K = 8
I chose 8 and not 27 because want the preprossesing to help the models not to be a model by itsef.

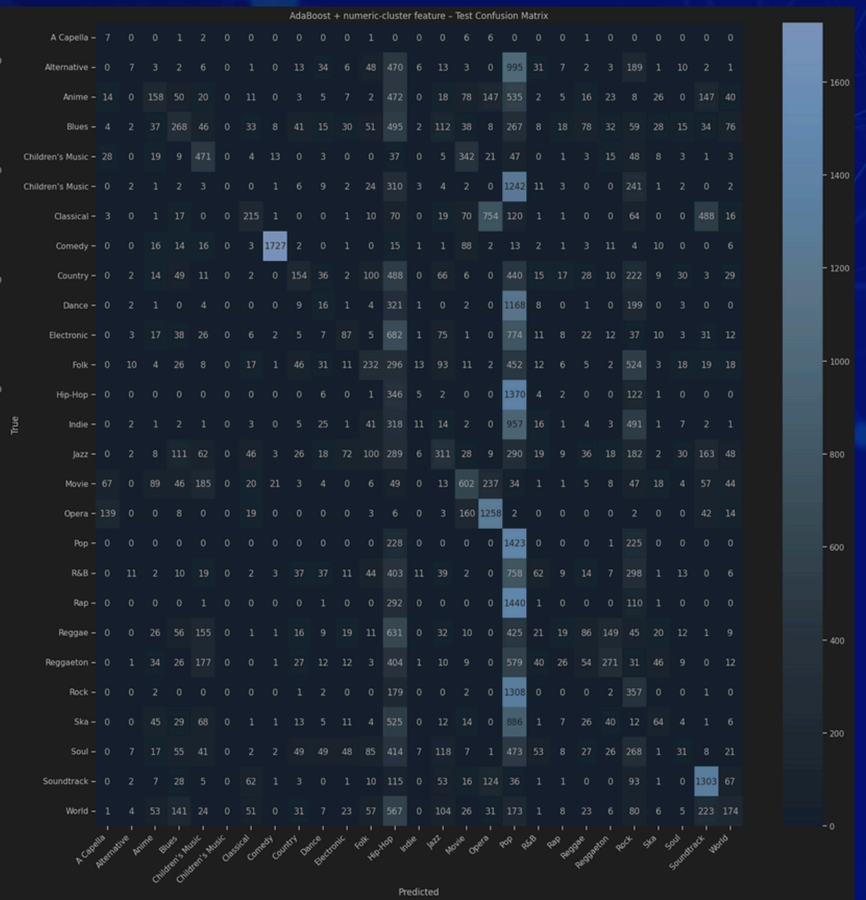
K-mean:
AdaBoost.
Random-Forest.

	acc	F1
RandomForest KMeans	41.1% /	41.9%
AdaBoost + KMeans	20.2% /	19.2%

Random-Forest



AdaBoost



The extra feature improves the performance slightly.

Summery

LightGBM	42.5% / 42.7%
Random-Forest + KMeans feature	41.1% / 41.9%
Random-Forest baseline	41.0% / 41.5%
Random-Forest + PCA	34.0% / 33.7%
AdaBoost	20.4% / 19.0%
AdaBoost + KMeans feature	20.2% / 19.2%
AdaBoost + PCA	17.3% / 14.5%

Thanks!

It was a pleasure to work on this project.