Music Genre Classification

KAGGLE COMPETITION ORGANIZED BY SHAI FOR AI

Lujain Moualla Muhammad Hassan Kinan Hasan

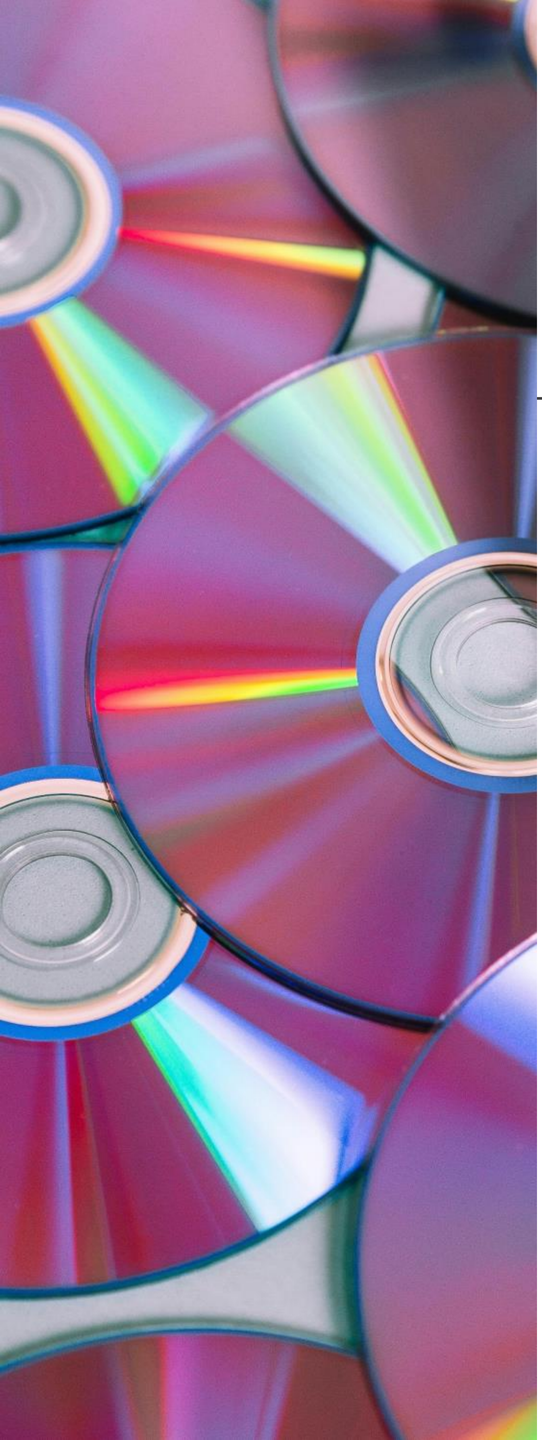


Table of Content

Look at The Big Picture

Dataset Overview:

- **artist:** Name of the Artist.
- **song:** Name of the Track.
- **popularity:** The higher the value the more popular the song is.
- **danceability:** Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm.
- **energy:** Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity.
- **key:** The key the track is in. Integers map to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on..
- **loudness:** The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative
- **mode:** Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
- **speechiness:** Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.

Dataset Overview:

- **acousticness:** A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
- **instrumentalness:** Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.
- **liveness:** Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.
- **valence:** A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).
- **tempo:** The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.
- **duration in milliseconds :**Time of the song.
- **time_signature :** a notational convention used in Western musical notation to specify how many beats (pulses) are contained in each measure (bar), and which note value is equivalent to a beat.
- **Class:** Genre of the track.

Dataset Information:

- The dataset consists of 17.995 entries and 18 columns.
- The columns are mostly of type float64, with some being int64 and object.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17996 entries, 0 to 17995
Data columns (total 18 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Id                  17996 non-null  int64  
 1   Artist Name         17996 non-null  object 
 2   Track Name          17996 non-null  object 
 3   Popularity          17568 non-null  float64
 4   danceability        17996 non-null  float64
 5   energy              17996 non-null  float64
 6   key                 15982 non-null  float64
 7   loudness            17996 non-null  float64
 8   mode                17996 non-null  int64  
 9   speechiness         17996 non-null  float64
 10  acousticness        17996 non-null  float64
 11  instrumentalness    13619 non-null  float64
 12  liveness            17996 non-null  float64
 13  valence             17996 non-null  float64
 14  tempo               17996 non-null  float64
 15  duration_in min/ms  17996 non-null  float64
 16  time_signature      17996 non-null  int64  
 17  Class               17996 non-null  int64  
dtypes: float64(12), int64(4), object(2)
memory usage: 2.5+ MB
```

Dataset **Information:**

more information about **String features**

1.Artist Diversity: There is a significant diversity in the artists represented, with 9149 unique artists.

2.Popularity: Backstreet Boys are the most frequent artist in this dataset, it appears 69 times indicating their high popularity.

3.Track Diversity: The dataset contains a wide variety of tracks, with 15129 unique track names.

4.Most Common Track: The track "Dreams" is the most common, but it only appears 9 times, suggesting a high variety of tracks overall.

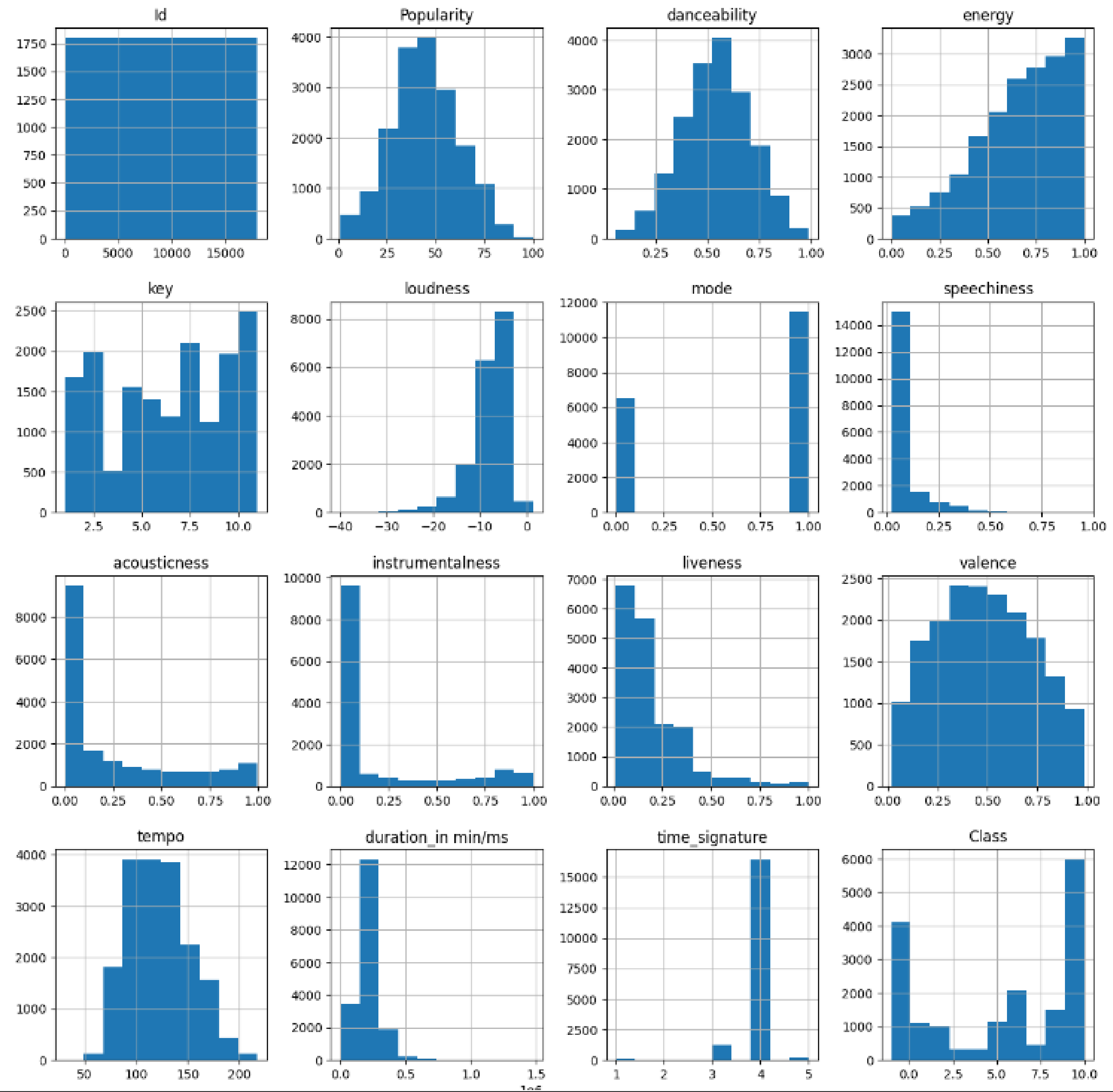
	Artist Name	Track Name
count	17996	17996
unique	9149	15129
top	Backstreet Boys	Dreams
freq	69	9

Dataset **Description:**

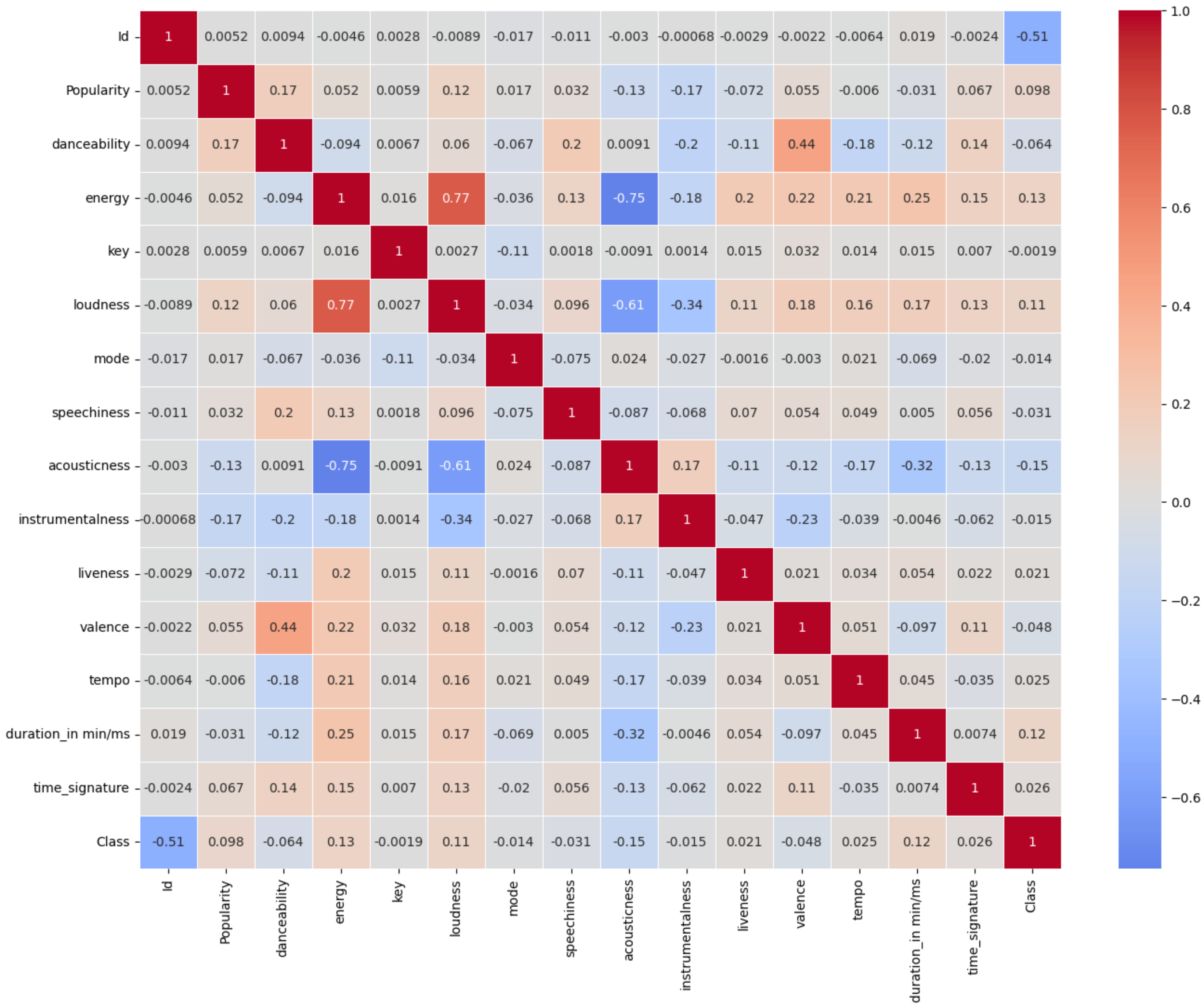
	count	mean	std	min	25%	50%	75%	max
Id	17996.0	8998.500000	5195.142058	1.000000	4499.750000	8998.50000	13497.25000	17996.000
Popularity	17568.0	44.512124	17.426928	1.000000	33.000000	44.00000	56.00000	100.000
danceability	17996.0	0.543433	0.166268	0.059600	0.432000	0.54500	0.65900	0.989
energy	17996.0	0.662777	0.235373	0.000020	0.509000	0.70000	0.86000	1.000
key	15982.0	5.952447	3.196854	1.000000	3.000000	6.00000	9.00000	11.000
loudness	17996.0	-7.910660	4.049151	-39.952000	-9.538000	-7.01600	-5.18900	1.355
mode	17996.0	0.636753	0.480949	0.000000	0.000000	1.00000	1.00000	1.000
speechiness	17996.0	0.079707	0.083576	0.022500	0.034800	0.04740	0.08300	0.955
acousticness	17996.0	0.247082	0.310632	0.000000	0.004300	0.08140	0.43400	0.996
instrumentalness	13619.0	0.177562	0.304048	0.000001	0.000089	0.00391	0.20000	0.996
liveness	17996.0	0.196170	0.159212	0.011900	0.097500	0.12900	0.25800	1.000
valence	17996.0	0.486208	0.240195	0.018300	0.297000	0.48100	0.67200	0.986
tempo	17996.0	122.623294	29.571527	30.557000	99.620750	120.06550	141.96925	217.416
duration_in min/ms	17996.0	200744.458851	111989.127131	0.501650	166337.000000	209160.00000	252490.00000	1477187.000
time_signature	17996.0	3.924039	0.361618	1.000000	4.000000	4.00000	4.00000	5.000
Class	17996.0	5.156201	4.207245	-1.000000	1.000000	6.00000	9.00000	10.000

Explore and Visualize Data to Gain Insights

Histogram plot



Correlation
Matrix



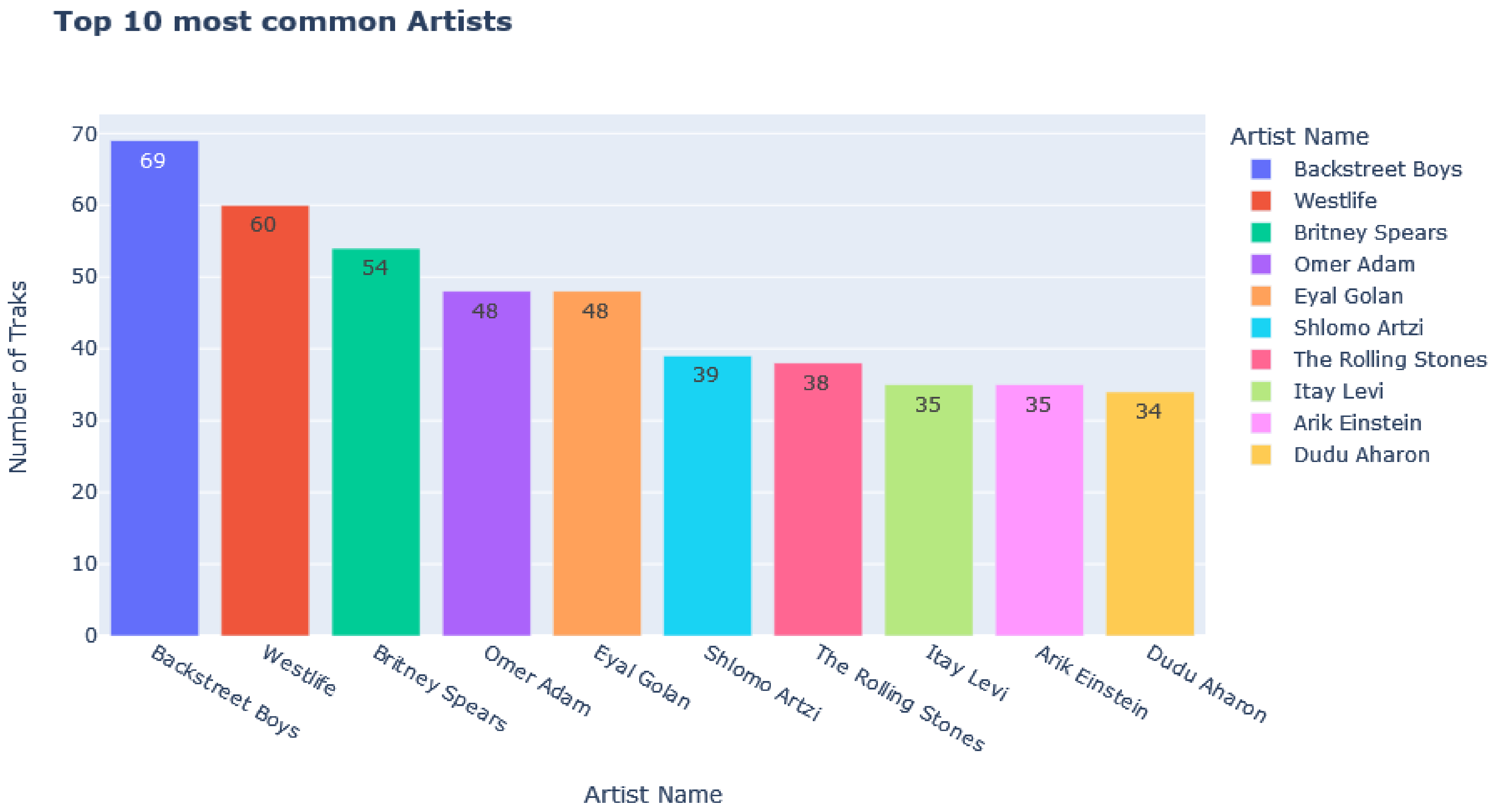
Correlation With 'Class'

there is no strong linear relationship between the features and the label.

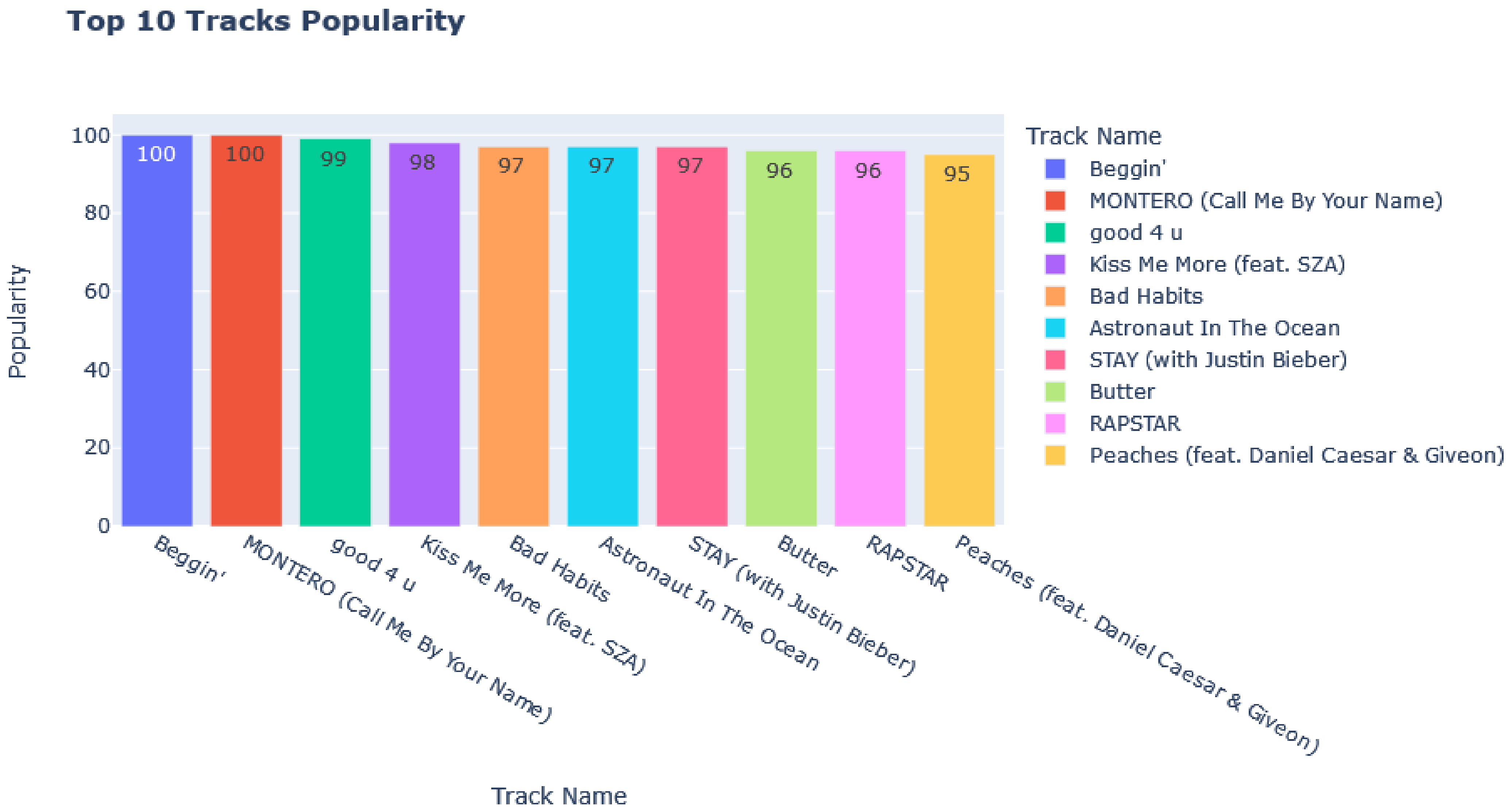
	Class
Class	1.000000
energy	0.129562
duration_in min/ms	0.115422
loudness	0.109899
Popularity	0.097927
time_signature	0.025551
tempo	0.024572
liveness	0.020890
key	-0.001932
mode	-0.013757
instrumentalness	-0.015004
speechiness	-0.030704
valence	-0.047605
danceability	-0.064270
acousticness	-0.148396
Id	-0.505689

dtype: float64

Top 10 Artists

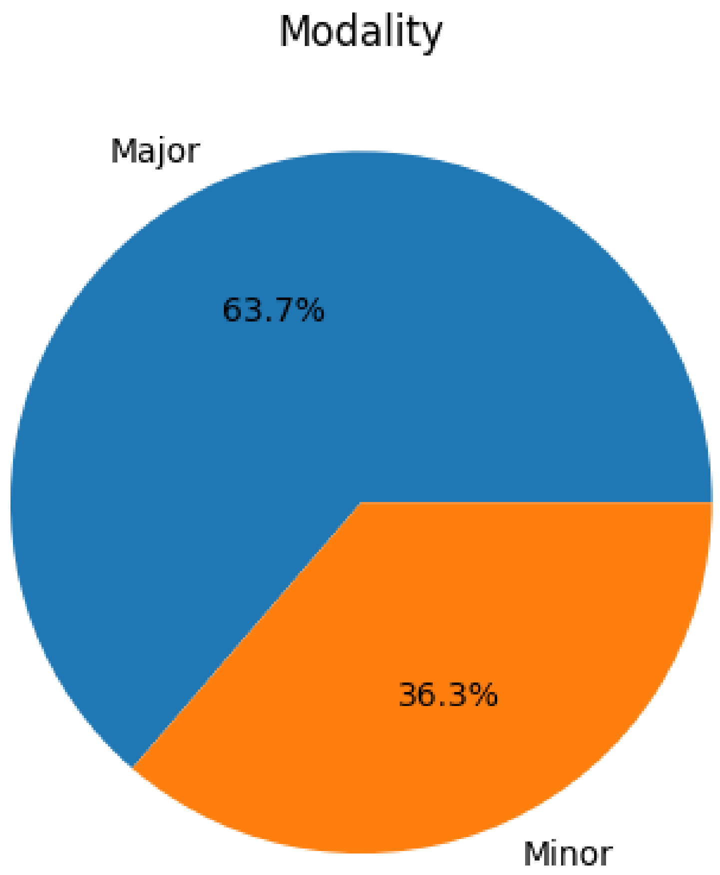


Top 10 Tracks Popularity



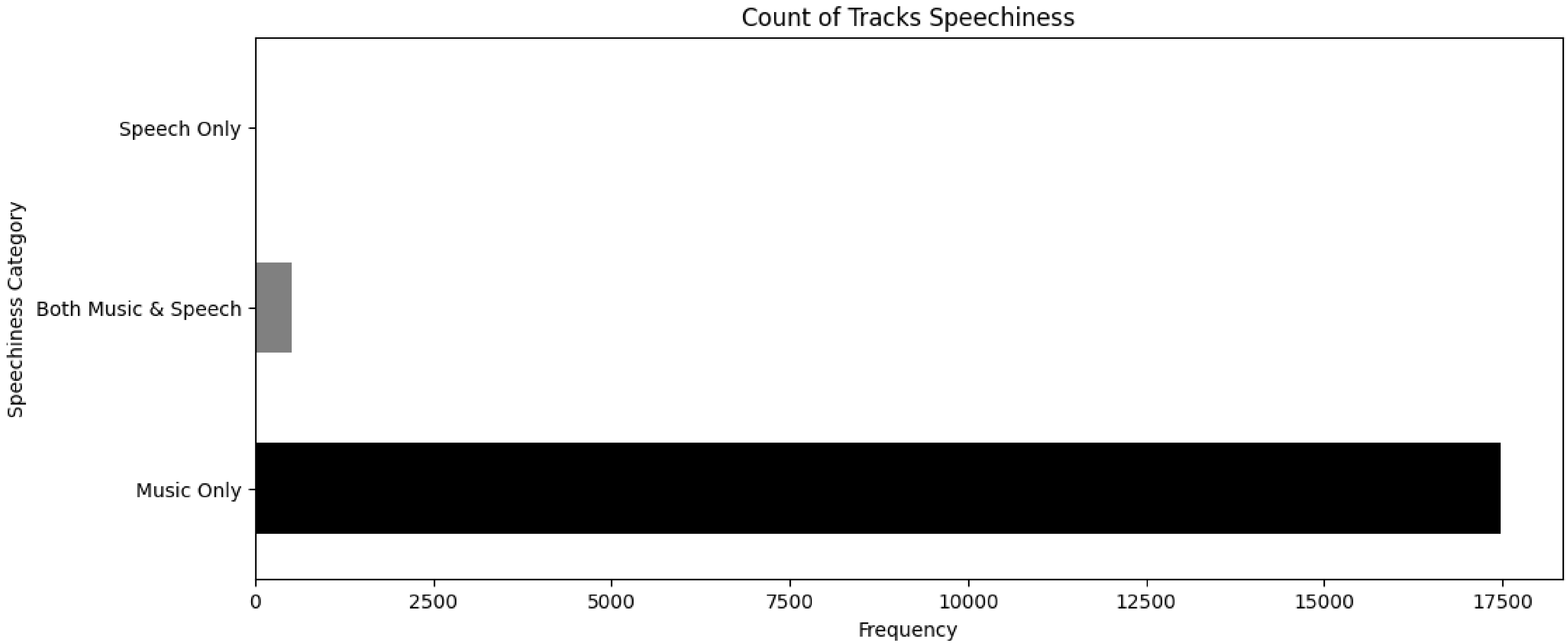
Modality

the majority of tracks in the dataset are in a major key (63.7%), while minor key tracks make up a smaller proportion (36.3%). This suggests a general preference for major keys in the dataset.



Count of Tracks Speechiness

The vast majority of tracks in the dataset are classified as "Music Only," indicating a low presence of spoken words. This suggests that the dataset primarily consists of instrumental or vocal music with minimal spoken content.

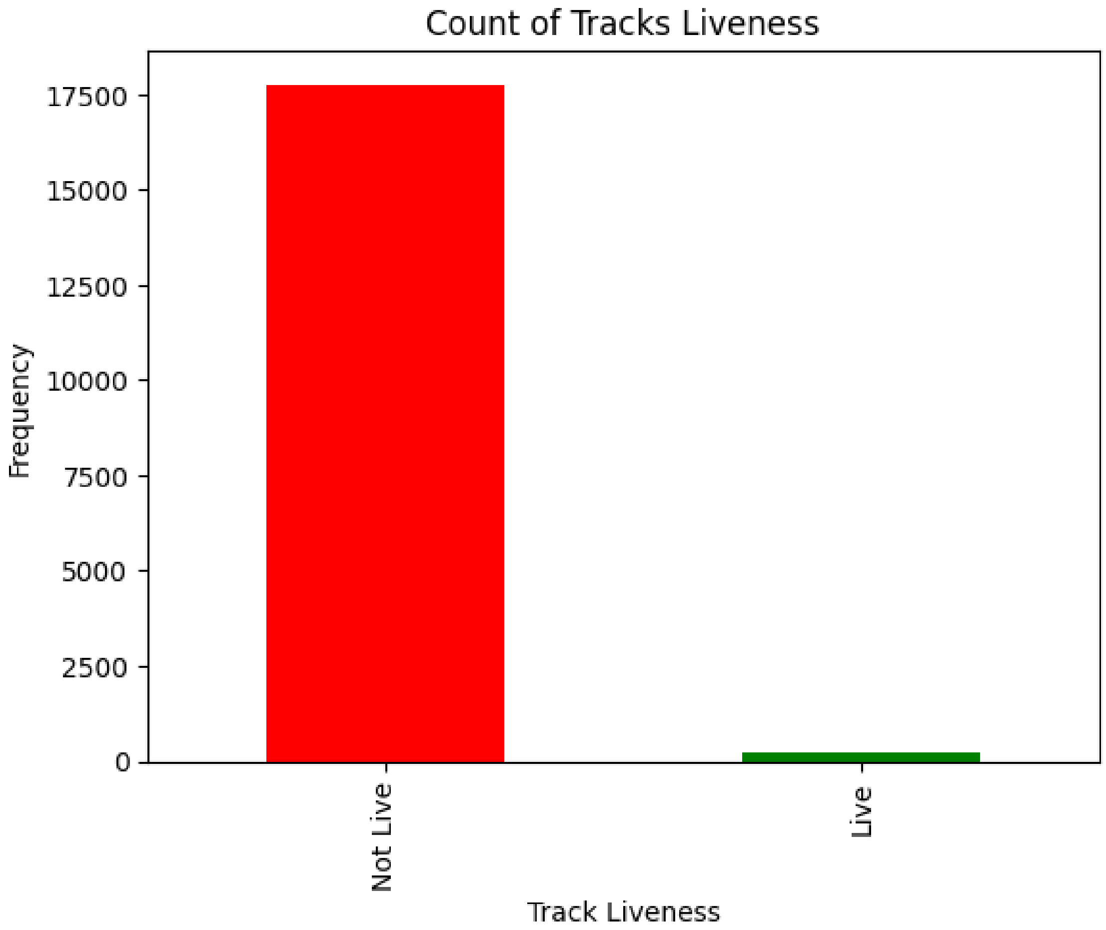


Count of Tracks Liveness

The vast majority of tracks in the dataset are classified as "Not Live".

This indicates a low likelihood of the tracks being recorded in a live setting.

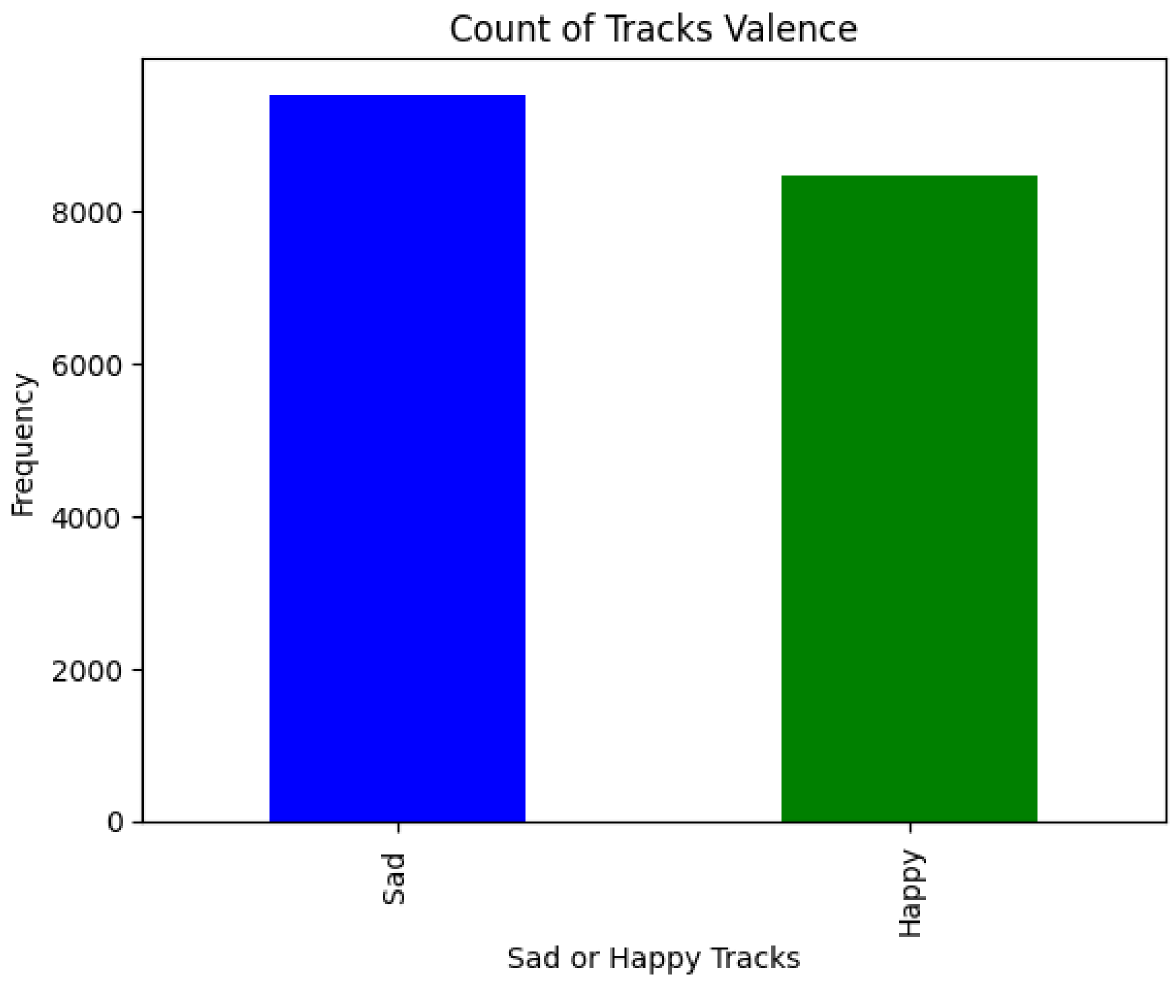
This suggests that the dataset predominantly consists of studio recordings rather than live performances.



Count of Tracks Valence

The distribution of valence, which measures the musical positiveness of a track, is almost balanced between "Sad" and "Happy" categories.

This suggests that the dataset contains a diverse range of emotions, offering a mix of tracks that evoke both positive and negative feelings.



Feature Engineering

Feature Extraction

Converting the ‘duration_in min/ms’ column into min :

-The `dur` variable stores the values from the `duration_in min/ms` column of the DataFrame.

-Two empty lists, `track` and `clean`, are initialized to hold processed data.

The loop iterates over each value in the `dur` list.
- If the value is less than or equal to 100:
    - The value is appended to `clean` without any modification.
    - The `track` list is appended with `0`, indicating that the value is in minutes.

Feature Extraction

If the value is greater than 100 :

-The value is assumed to be in milliseconds, so it's divided by 60,000 to convert it into minutes.

-The `track` list is appended with `1`, indicating that the value was in milliseconds and was converted to minutes.

-The original `duration_in min/ms` column in Data frame is updated with the values from `clean`, meaning all the values are now consistently in minutes.

- A new column `new1` is added to the DataFrame. This column indicates the original unit of the duration (`0` for minutes, `1` for milliseconds).

Feature Engineering

-Importance in a Classifier:

Consistency : Ensures all inputs are consistent, which is crucial for accurate predictions

1.`new2`: Length of the 'Artist Name' (number of characters).

2. `new3`: Length of the 'Track Name' (number of characters).

3. `new4`: Number of words in the 'Artist Name' (based on splitting the string by spaces).

4. `new5`: Number of words in the 'Track Name’.

5. `new6`: Number of uppercase letters in the 'Artist Name'.

Feature Engineering

6. `new7`: Number of lowercase letters in the 'Artist Name'.
7. `new8`: Number of digits in the 'Artist Name'.
8. `new9`: Number of uppercase letters in the 'Track Name'.
9. `new10`: Number of lowercase letters in the 'Track Name'.
10. `new11`: Number of digits in the 'Track Name'.
11. `new12`: Count of special characters in the 'Artist Name' (not uppercase, lowercase, or digits). Calculated by subtracting the sum of uppercase and lowercase counts from the total length of the 'Artist Name'.
12. `new13`: Count of special characters in the 'Track Name' (not uppercase, lowercase, or digits). Similar logic as `new12`, applied to 'Track Name'.

Why These Features Might Be Useful ?

Text Characteristics as Predictors:
In a music genre classification task, the characteristics of the `Artist Name` and `Track Name` might provide hints about the genre. For example:
Length of Name: Certain genres might have trends in artist or track name lengths.
Uppercase/Lowercase Letters: Some genres might favor stylized names with more uppercase letters or special characters.
Digits in Names: Genres like EDM or hip-hop might use numbers more frequently in track or artist names.

Why These Features Might Be Useful ?

-Model Input: - These features can be fed into a machine learning model to help it learn patterns associated with different music genres, improving its predictive power.

*Example Use Case

-Pop Music: Shorter, simpler track names, potentially fewer special characters.

- Hip-Hop: More use of digits, special characters, and stylized names with uppercase letters.
- Classical: Longer artist names, possibly fewer special characters.

Feature Engineering

-Standardize the text by converting it to lowercase:

Why??

1. Case Insensitivity:
   - Converting text to lowercase is a common preprocessing step in text data to make the data case-insensitive. This ensures that variations in capitalization (e.g., "The Beatles" vs. "the beatles") are not treated as different entities.

2. Consistency in Feature Engineering:
   - By converting all text to lowercase, subsequent feature engineering processes, such as counting specific characters, words, or patterns, will yield consistent results.

3. Improved Model Performance:
   - Machine learning models often perform better when the input data is consistent and standardized. By cleaning the text, the model can focus on the meaningful patterns in the data rather than getting distracted by irrelevant variations like case differences.

Model Training

AutoGluon

Model Training:

We Have Used AutoGluon to select our machine learning model.

AutoGluon Configurations:

Best Quality: Automatically selects the best-performing model.

Auto Stack: Enhances model performance through automated stacking.

Final Model:

- **The Best Model:**
  - Weighted Ensemble of the best models with l2 regularization to avoid overfitting

- **Final F1 Score:**
  - .74718

	model
1	WeightedEnsemble_L2
2	XGBoost_BAG_L1
3	NeuralNetFastAI_BAG_L1
4	LightGBMXT_BAG_L1
5	LightGBM_BAG_L1
6	ExtraTreesEntr_BAG_L1
7	RandomForestEntr_BAG_L1
8	RandomForestGini_BAG_L1
9	ExtraTreesGini_BAG_L1
10	KNeighborsDist_BAG_L1

Final Result:

Music Genre Classification 2024

Late Submission ···

Overview | Data | Code | Models | Discussion | Leaderboard | Rules | Team | Submissions

Search leaderboard

Public | Private

The private leaderboard is calculated with approximately 30% of the test data.
This competition has completed. This leaderboard reflects the final standings.

#	△	Team	Members	Score	Entries	Last	Solution
1	—	Samer Emad Harb		0.83983	70	8d	
2	▲ 1	**Team 2**		0.74718	13	6d	

The End