CSE343 Machine Learning - Interim Project Report

Manav Mittal
2021538
manav21538@iiitd.ac.in

Utkarsh Venaik
2021570
utkarsh21570@iiitd.ac.in

Akash Kushwaha
2021514
akash21514@iiitd.ac.in

Lakshay Kumar
2021061
lakshay21061@iiitd.ac.in

1. Motivation

We all love music, don't we? Most of us might also have some favorite songs, some songs we just genuinely like, and some songs we simply dislike. We usually think that it is simply by coincidence that we love some songs while disliking others but what we are trying to study from this project is whether or not there is an actual data science aspect hidden behind certain songs being more popular and loved than others. What we will be trying to achieve in this project is to use characteristics of music to predict whether or not it will be popular among the listeners and also the degree of popularity that it may achieve. This study can also be useful for music composers, singers, and instrumentalists in deciding what their new music should sound like and whether or not it will be loved by the audience. All the team members thought this is a very interesting and innovative topic to build a project upon and there is also a lot of scope for innovation in this field since music and its characteristics can be studied and defined in terms of many factors and characteristics. We also thought that as we grow our skills in the field of machine learning and processing of waves and signals there will be a scope for a lot of improvement and various more would be opening up to dig deeper into this topic hence this would be a great project to work and iterate upon in the future as well for all the team members.

2. Introduction

This project focuses on analysing if there exists a relationship between music popularity and song characteristics. This project dives deeper into trying to answer the question of whether a particular song's popularity has some underlying data-driven factors leading to it. The main objective of this project is to leverage attributes of a song to predict its popularity and also predict the degree of popularity it may achieve.

3. Literature Review

3.1. Reference 5

In this paper, they are basically building a methodology that can predict whether a song will appear on Spotify's Top 50 Global ranking after a certain amount of time. They approach the problem as a classification task and use the data from the past platform's Top 50 Global ranking collected using Spotify's web API, The model uses information on the songs previously observed in that list, Support Vector Machine classifier with RBF kernel reached the best results in our experiments with an AUC higher than 80% when predicting the popularity of a song two months in advance.
[Presented at Conference: Simpósio Brasileiro de Computação Musical]

3.2. Reference 6

This project focuses on predicting the popularity of songs on the Million Song Dataset, a crucial aspect for maintaining competitiveness in the expanding music industry. The dataset containing audio features and metadata for around one million songs, the study assesses various classification and regression algorithms to determine their effectiveness in forecasting song popularity. The investigation also identifies the specific types of features that possess the greatest predictive capability in this context.

In this paper, they evaluated different classifications and regression algorithms on their ability to predict popularity and determined the types of features that hold the most predictive power.
[Presented at Conference: ACM]

3.3. Reference 7

This paper primarily focuses on the Popularity Prediction of Music by related Python tools and various machine learning models like xgboost, XGBRegressor, and Polynomial Regression, the dataset for the study contains

114000 data points and 19 features. The research evaluates models using metrics such as mean-squared error and R-squared. Ultimately, XGBoost emerges as the best-fitting model, demonstrating a strong correlation between music attributes and popularity. The paper also discusses potential areas for improvement, including refining categorical variable encoding and exploring interactions between independent variables.
[Presented at Conference: IEEE]

3.4. Mathematical Concepts

Mean: It is the total sum of values in the dataset divided by the total number of data points, it is a measure of the central tendency of a probability distribution along median and mode.
Standard deviation: It is the measure of the spread or dispersion of a set of data points from their mean, It helps assess the consistency and reliability of data which in turn draws meaningful conclusions.
Covariance: It is a measure of the relationship between two random variables and to what extent, it indicates whether an increase in one variable corresponds to an increase or decrease in another variable. A positive covariance suggests a positive relationship, while a negative covariance indicates an inverse relationship and a covariance of zero implies no linear relationship between the variables.
Correlation: It is a statistical measure that expresses the extent to which two variables are linearly related.

4. Dataset Description

The primary source of our dataset is the Spotify API which analyses the songs uploaded on its platform on the basis of various sound characteristics. The Spotify API essentially provides us with 2 useful datasets, one which primarily contains the numerical parameters such as:

1. song_duration_ms: This refers to the length of the track in milliseconds.
2. acousticness: This is a confidence measure from 0.0 to 1.0 that assesses whether the track is acoustic. A value of 1.0 indicates high confidence that the track is acoustic.
3. danceability: This describes how suitable a track is for dancing, considering elements like tempo, rhythm stability, beat strength, and overall regularity. Least danceable is 0.0 to 1.0 being most danceable.
4. energy: This is a measure from 0.0 to 1.0 that represents the perceptual intensity and activity of the track. Energetic tracks typically feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale.
5. instrumentalness: Detects the number of different instruments involved in the songs.
6. key: This represents the estimated overall key of the track, with integers mapping to pitches using standard Pitch Class notation.
7. liveness: Detects the presence of an audience in the recording. Value is proportional to probability that music was recorded in a live event
8. loudness: Measured in decibels (dB). It is basically the decibel value for each instance averaged across the entire track and are useful for comparing relative loudness of tracks. Values typical range between -60 and 0 db.
9. audio_mode: This indicates the modality (major or minor) of the track, the type of scale from which its melodic content is derived. Major is represented by 1, and minor is represented by 0.
10. speechiness: Measure of the presence of spoken words in a track. When the track is something like an audiobook or podcast where there is mostly clearly said and complete words with a lot of understanding and speech involved this parameter tends to 1 and the other part of the spectrum is the songs which have absolutely no wordings involved in them.
11. tempo: beats per minute (BPM). i.e. the pace of a given song
12. time_signature: This provides an estimated overall time signature of the track, specifying how many beats are in each bar (or measure).
13. audio_valence: range: [0, 1] is a measure of the musical positiveness conveyed by a track. Higher value means happier the song is while lower the value means that the song is sad, intermediate values convey the balance between positivity and negativity of the song
14. song_popularity: Based on rating and number of listens and re-listens by the audience.

This dataset is the song_data.csv, other than this there is another dataset for some additional information such as:

1. artist_name: is the name of the singer of the track
2. album_names
3. playlist: is the playlist name in which the song is launched in spotify.
4. song_name

This dataset is named song_info.csv

For our project, we started off by making a new dataset using their primarily available datasets. To form our dataset we have used the concat() function horizontally ie: on axis1 to both the datasets, since the artist_name column is common in both of them we dropped it from the second dataset, while since our target variable is the song popularity we added it as the target column towards the end.

Final dataset dimensions: (18835, 19)

This is followed by classifying the numerical score into categories to turn it into a bounded classification problem:

Category	Popularity Score
Potential Masterpiece	78-100
Popular	69-78
Mildly Popular	56-69
Potential Flop	0-56

Table 1. Category wise popularity score

These boundaries have been decided based upon certain percentile calculations which has been discussed more in detail further.

Category	Percentile
Potential Masterpiece	90%
Popular	75-90%
Mildly Popular	50-75%
Potential Flop	0-50%

Table 2. Category wise percentile

We found out what are the percentile scores for each and also plotted distribution plots and quartiles to decide on the numbers

Field	Value
mean	52.991877
std	21.905654
min	0.000000
5%	8.000000
10%	21.000000
25%	40.000000
50%	56.000000
75%	69.000000
80%	72.000000
90%	78.000000
95%	85.000000
max	100.000000

Table 3. Caption

Other than this we have done the Standardization of the dataset to improve the model predictions and accuracy later on however minor tweaking is expected to be needed as we move on further into the project. Encoding was needed for the varchar values in the dataset ie: song_name, artist_name, album_name, playlist we have used one-hot encoding for it. Missing values were printed however the dataset has no NaN or missing values hence not needed to be removed/replaced with the average. Distribution curve was plotted to find the percentile and get an idea of the distribution. The resultant plot is a normal distribution as expected. Feature selection and outlier detection has also been done which is been decsribed in more detail in the methodology.

5. EDA

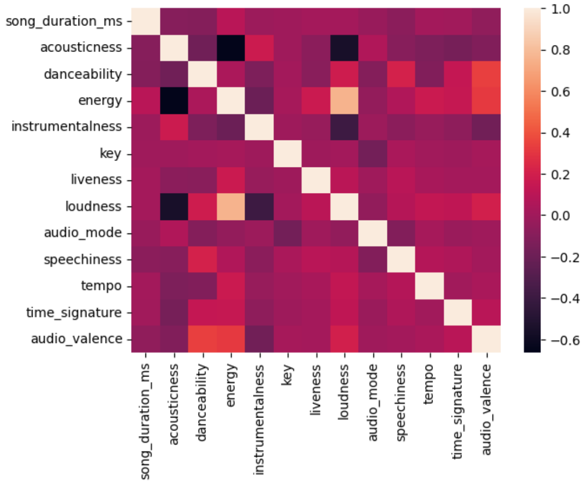


Figure 1. Heatmap of Features

Observations from Figure 1 are:

1. loudness is inversely correlated with number of instruments.
2. loudness is correlated with energy.
3. loudness is inversely correlated with acousticness.
4. acousticness is inversely correlated with energy.

We have performed a very detailed EDA for every feature to see that please refer to annexure.

6. Methodology and Model description

6.1. Data Collection

We obtained two primary data files, namely ’song_data.csv’ and ’song_info.csv’, and details about these files are described above in the ‘Data description’ section.

6.2. Data Integration:

We merged these two datasets using the Python pandas library to create a unified dataset.

6.3. Categorization of Songs

To categorize the songs effectively, we utilized the describe function to calculate percentiles and scores. We classified the songs into four distinct categories based on their popularity score:

- **Potential Masterpiece:** This category comprises songs with exceptionally high scores, indicating significant potential for success.
- **Popular:** These are songs that have already achieved a high level of popularity.
- **Mildly Popular:** This category includes songs with moderate levels of popularity.
- **Potential Flop:** Songs in this category have scores suggesting a lower likelihood of success.

For categorization of data in described categories we computed the percentage of data falling into 4 categories:

- Percentage of songs that are potential masterpieces: 10.96%
- Percentage of songs that are popular: 14.57%
- Percentage of songs that are mildly popular: 24.48%
- Percentage of songs that are potential flops: 49.97%

6.4. Data Pre-processing

We conducted initial data pre-processing, which involved: Checking for Null or Missing Values: We systematically examined the dataset for any missing or null values. We did min-max scaling on the dataset.

6.4.1 Feature Engineering

Since we decided to only have the song_name as the categorical data since the name of song can have impact on its popularity. First we created a bag of words then replace the song_name with the most occurring word in that name and then did one hot encoding on the song_name after that we ran PCA on the one-hot encoded dataset and found that variance is too much distributed among the features. After encoding, features set became of size more than 6000 and 1000 features were incorporating about 60% of the variance in the data as can be seen in figure 2. As we can see number of features have increased too much computational complexity will increase exponentially.

So we look for any other method to do this and we decided to do sentiment analysis. In this case we used well known NLP library to judge the sentiment of the song just by looking at its name. We used TextBlob and NTLK to do the sentiment analysis. We found similar results in the cases of both and we just decided to stick with TextBlob. Based on sentiment analysis we assigned -1 (negative sentiment),0 (neutral sentiment) or 1 (positive sentiment) to data sample depicting its sentiment.

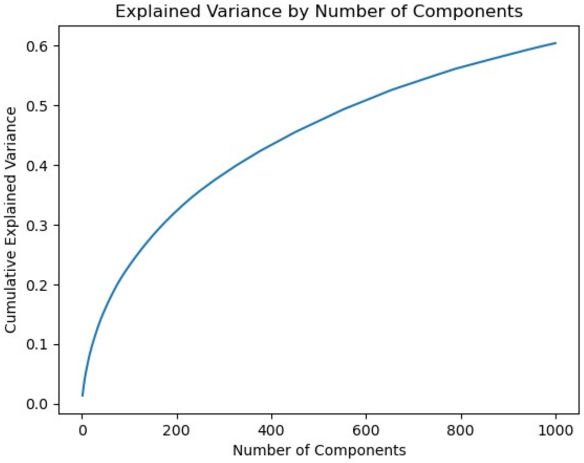


Figure 2. Explained variance vs. Number of components

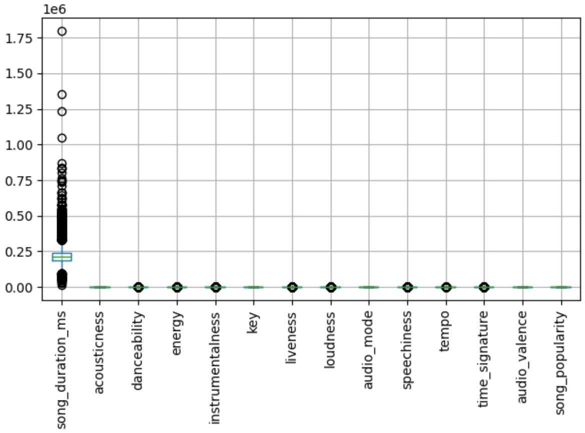


Figure 3. Boxplot of features (I)

6.4.2 Outlier Detection and Removal

We used a boxplot to detect the outliers in the data. From figure 3 and Figure 4 we can clearly see that there are many sample points in the dataset which are deviating from the central tendency of the data therefore there will be outliers in the dataset and we have to figure out some methodology to remove them.

We used Local Outlier Factor (LOF) score with 1% contamination to remove the outliers when we increased the contamination the accuracy was getting affected. Using this we removed around 180 data samples.

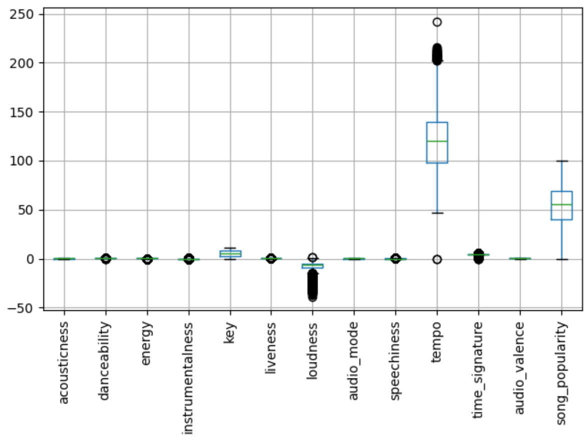


Figure 4. Boxplot of features (II)

6.5. Model description

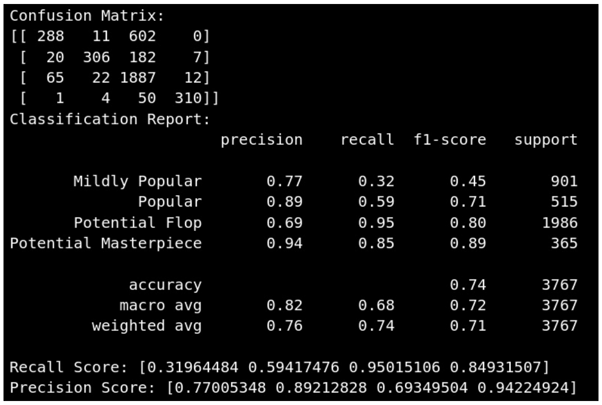
We have tried several classification models on the dataset. Following are the best accuracy scores we obtained after using grid search.

Model	Accuracy Score
Naive Bayes	0.511
Logistic Regression	0.523
Decision Tree	0.632
SVM (Linear)	0.530
SVM (RBF)	0.705
Random Forest	0.783
KNN	0.669
MLP	0.539
Ensemble (Random Forest and KNN)	0.713
AdaBoost	0.538

Table 4. Accuracy scores of different ML models

As we can see Random Forest performed best out of all the models so we further tried to improve the accuracy by applying the ensemble methods. We trained 3 different Random forests one with different subsets of 90 % features, one without bootstrapping and last one just with best parameters, no feature sampling and with bootstrapping we obtained using grid search and we also added KNN because we were getting fairly good accuracy by using it. But the overall accuracy reduced as we can see in the above table.

We found the random forest to be the best model. The best parameters for random forest we found are ['n_estimators': 1000, 'max_depth': None (Other parameters were default)].



```
Confusion Matrix:
[[ 288   11  602    0]
 [  20  306  182    7]
 [  65   22 1887   12]
 [   1    4   50  310]]
Classification Report:
                       precision    recall  f1-score   support

       Mildly Popular       0.77      0.32      0.45       901
              Popular       0.89      0.59      0.71       515
       Potential Flop       0.69      0.95      0.80      1986
Potential Masterpiece       0.94      0.85      0.89       365

             accuracy                           0.74      3767
            macro avg       0.82      0.68      0.72      3767
         weighted avg       0.76      0.74      0.71      3767


Recall Score: [0.31964484 0.59417476 0.95015106 0.84931507]
Precision Score: [0.77005348 0.89212828 0.69349504 0.94224924]
```

Figure 5. Confusion Matrix - Random Forest

7. Result and Analysis

After performing 10 K-fold cross validation tests on our Random Forest model we achieved a mean accuracy of almost 75% with a peak of 78% accuracy. The confusion matrix for the Random Forest model we created using the best parameters is shown in the fig 5.

8. Conclusion

In our quest to predict song popularity, this accuracy is a significant accomplishment in the intricate and subjective world of music. It is essential to understand that perfection is nearly impossible in music prediction due to the diverse and ever-changing nature of musical tastes. Our success in correctly predicting popularity in over three-quarters of cases underlines the model's robustness and the importance of sound characteristics in determining a song's appeal. This outcome is a validation of our approach and this model can be a valuable tool for artists and the industry, offering insights that go beyond intuition to a more data-driven understanding of what makes a song a hit.

9. References

1. https://www.kaggle.com/datasets/edalrami/19000-spotify-songs
2. https://towardsdatascience.com/song-popularity-predictor-1ef69735e380
3. https://developer.spotify.com/documentation/web-api
4. https://www.kaggle.com/code/amansorout/spotify-song-popularity-classification
5. https://www.researchgate.net/publication/341420234_

Predicting_Music_Popularity_on_Streaming_Platforms/link/5f0d0bd8a6fdcca32ae97ccc/download

6. http://cs229.stanford.edu/proj2015/140_report.pdf
7. https://www.researchgate.net/publication/370712842_Popularity_Prediction_of_Music_by_Machine_Learning_Models

10. Annexure

10.1. EDA

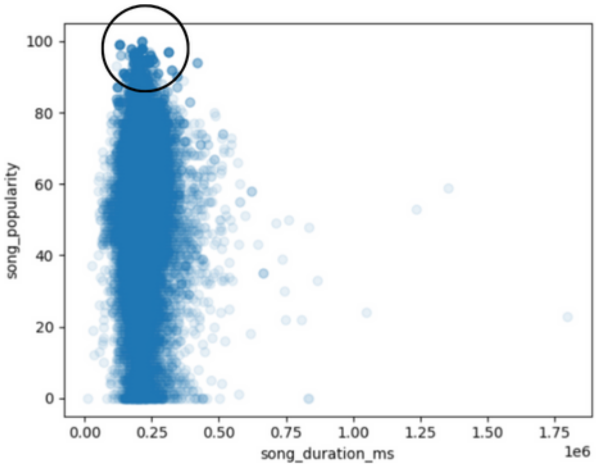


Figure 6. Song Popularity vs. Song Duration

Insights related to figure 4. are:

1. Most of the songs lie in a certain duration band which is less than 400 seconds.
2. Most of the popular songs are of duration around 250 +- 100 seconds.
3. The songs with very high duration are generally less likely to be popular and same is true with very low duration songs.
4. Since the score required to be a “masterpiece” is more than 85, from the plot it seems that if a song is way too much in duration it is not possible for it to be a masterpiece in terms of popularity.

In case of figure 5 we can observer that most of the songs in our dataset have acousticness on the lower side.

Insights related to figure 6. are

1. More danceable songs are more likely to have higher popularity.
2. Most of the songs in our dataset have danceability in the range of 0.3 to 0.9.

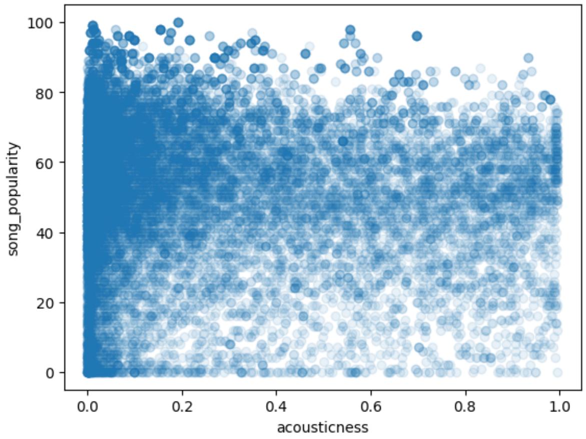


Figure 7. Song Popularity vs Acousticness

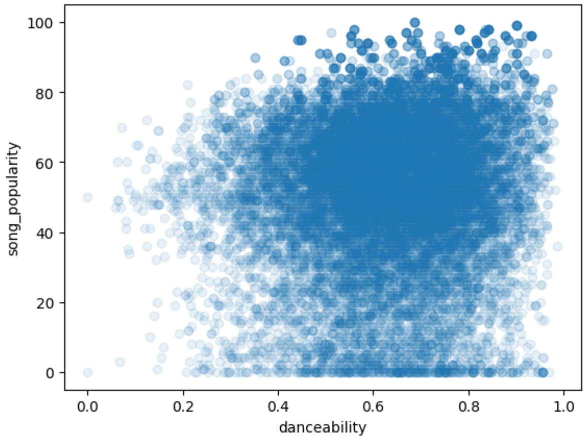


Figure 8. Song Popularity vs Danceability

Insights related to figure 7. are

1. The songs in our dataset generally have higher energy.
2. None of the overly energetic song Ie: with energy score greater than 0.95 seems to be a masterpiece.

Insights related to figure 8. are

1. Most of the songs have either very less number of instruments or too many number of instruments.
2. Almost all of the masterpieces use a very less number of different instruments.

Insights related to figure 9. are

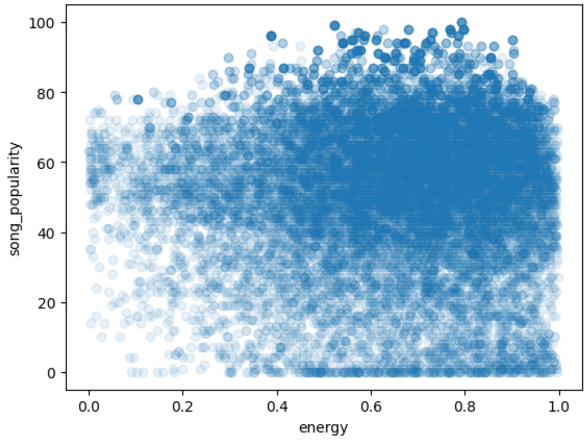


Figure 9. Song Popularity vs Energy

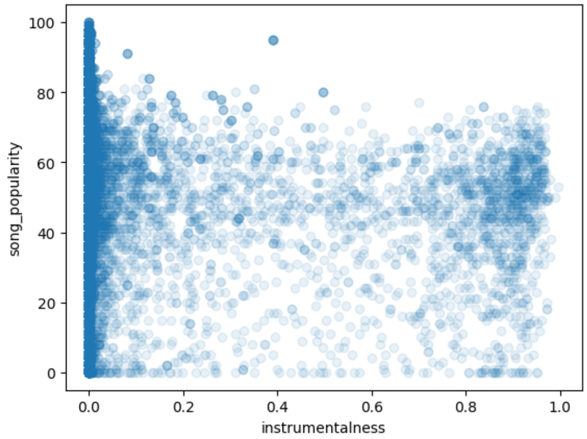


Figure 10. Song Popularity vs Instrumentliness

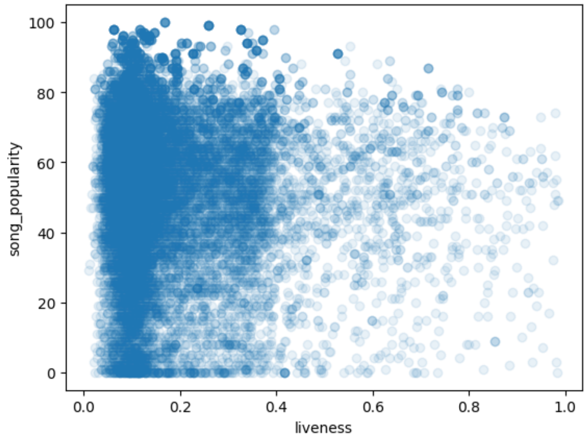


Figure 11. Song Popularity vs Liveliness

1. Most of the tracks have less liveliness.
2. Songs with less liveliness are likely to be very popular.

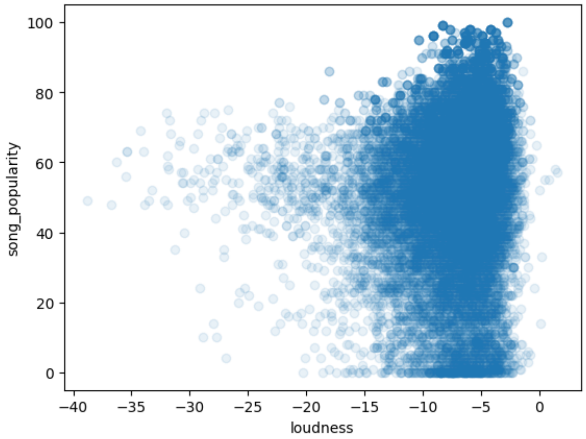


Figure 12. Song Popularity vs Loudness

From figure 10 we can observer that most of the songs generally are on the louder side.

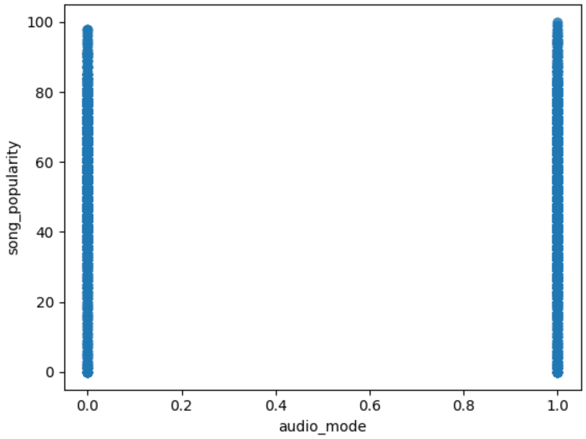


Figure 13. Song Popularity vs Audio Mode

From figure 11 we can observe that audio mode, individually does not seem to have a considerable relation with song popularity since both the audio modes have almost identical popularity distribution.

From figure 12 we can observer that songs generally tend to have less speechiness.

Insights related to figure 13. are

1. Tempo of songs generally lie in a concentrated band of values.
2. No songs have too high or too low tempo.

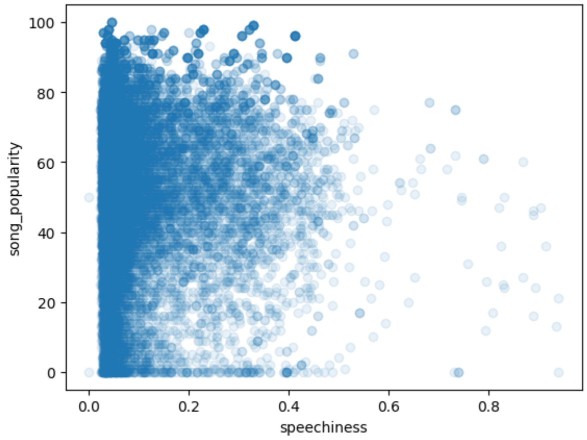


Figure 14. Song Popularity vs Speechiness

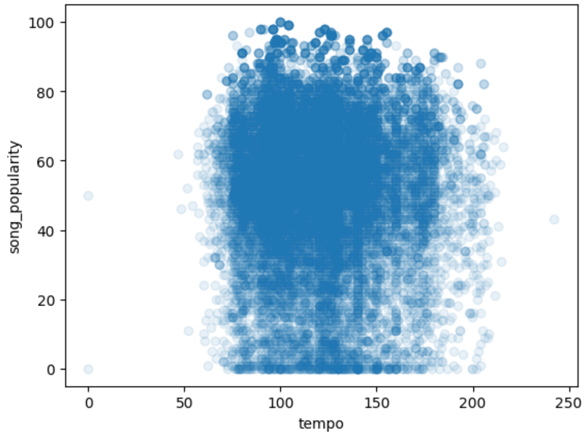


Figure 15. Song Popularity vs Tempo

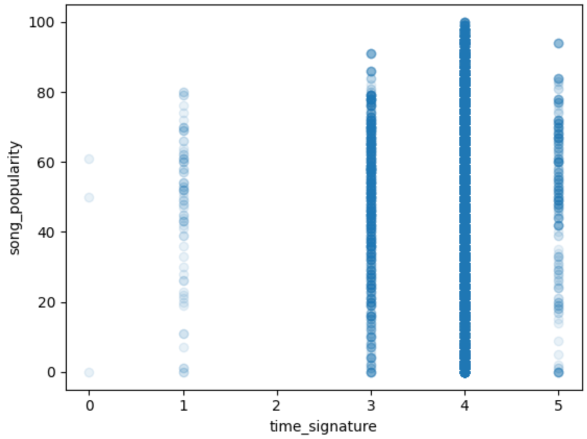


Figure 16. Song Popularity vs Time Signature

From figure 14 we can observe that most of the songs have time signature value of 3, 4 or 5.

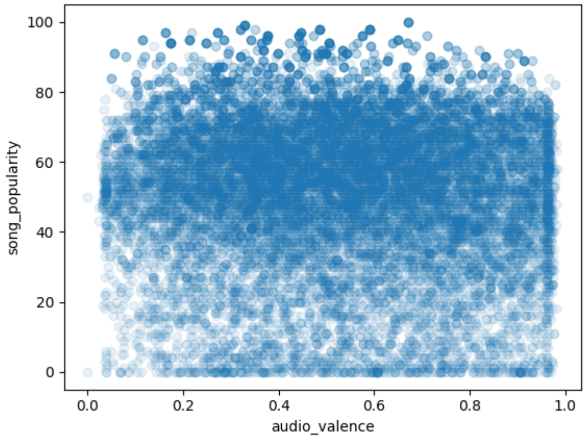
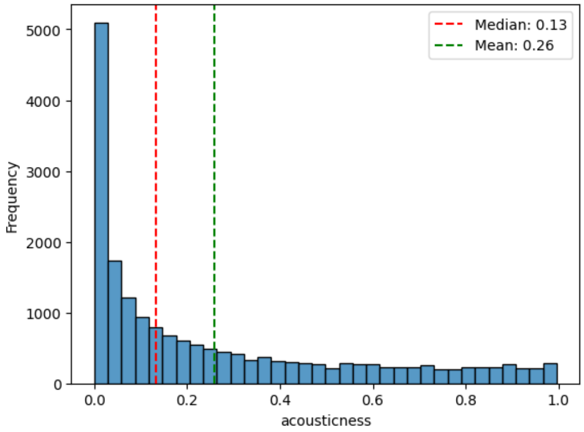
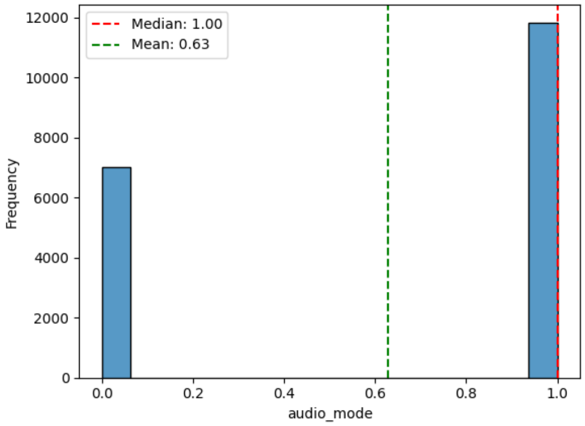


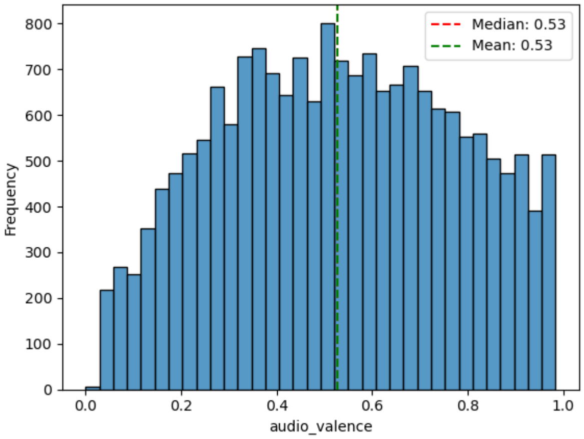
Figure 17. Song Popularity vs Audio Valence

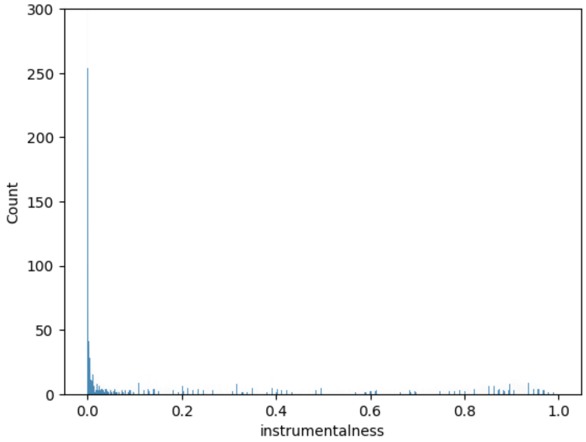
From figure 15 we can observer that every kind of song has its own good number of listeners ie: Every song either happy or sad is equally likely to be popular or flop.

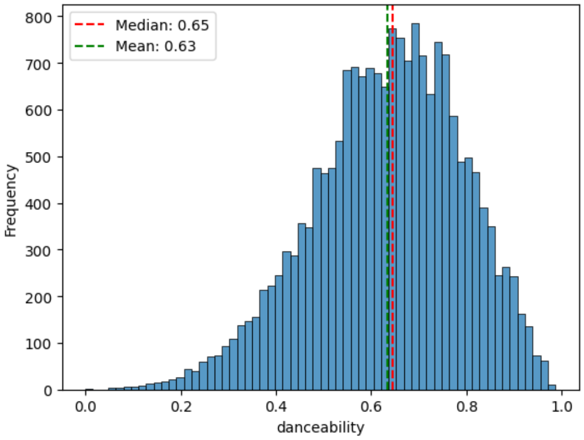
10.2. Some other relevant graphs

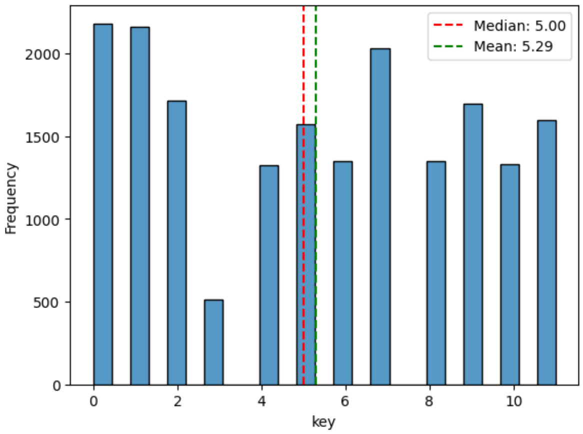


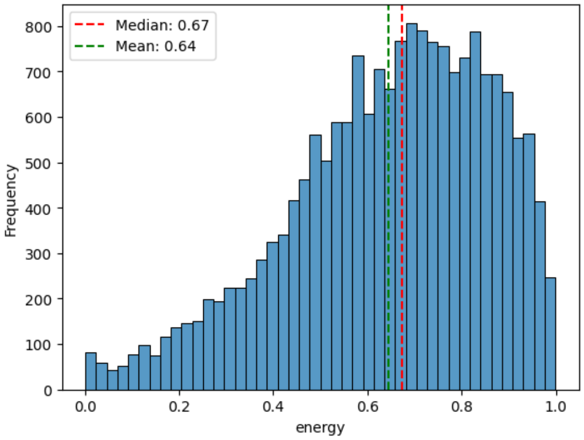


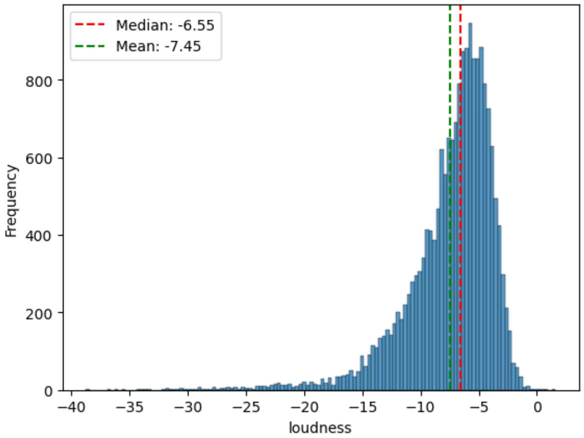

Median: 0.53
Mean: 0.53
Frequency
0
100
200
300
400
500
600
700
800
0.0
0.2
0.4
0.6
0.8
1.0
audio_valence

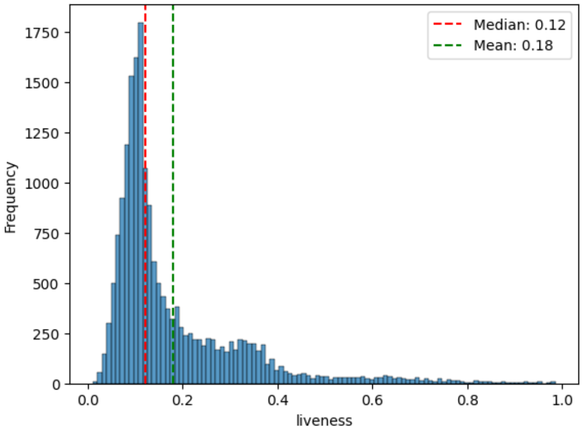

Count
0
50
100
150
200
250
300
0.0
0.2
0.4
0.6
0.8
1.0
instrumentalness

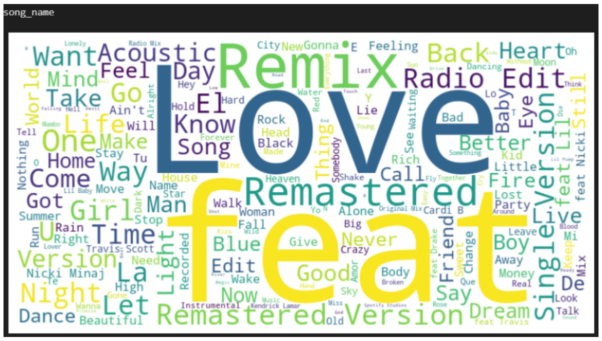

Median: 0.65
Mean: 0.63
Frequency
0
100
200
300
400
500
600
700
800
0.0
0.2
0.4
0.6
0.8
1.0
danceability

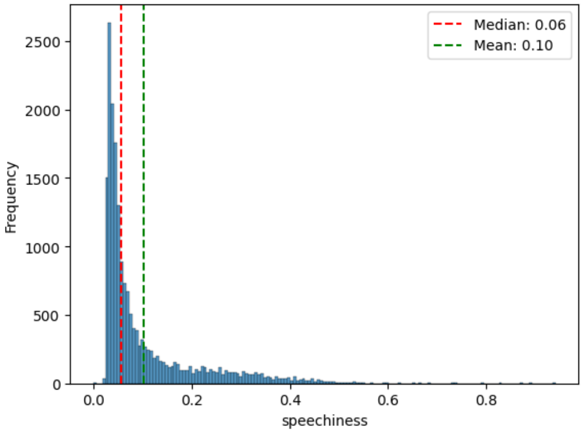

Median: 5.00
Mean: 5.29
Frequency
0
500
1000
1500
2000
0
2
4
6
8
10
key

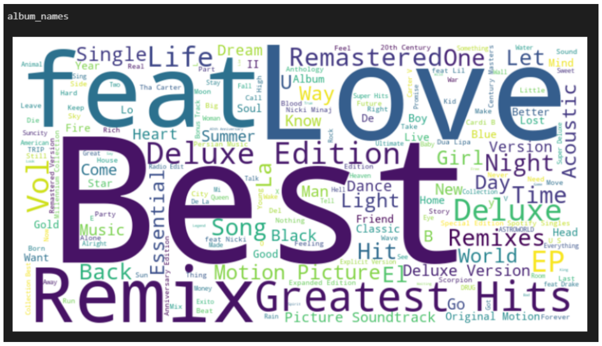

Median: 0.67
Mean: 0.64
Frequency
0
100
200
300
400
500
600
700
800
0.0
0.2
0.4
0.6
0.8
1.0
energy

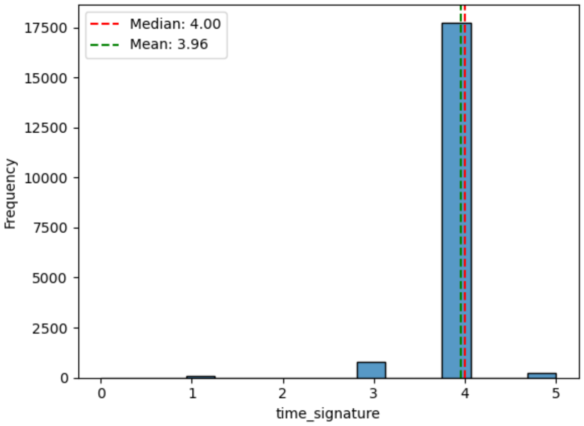

Median: -6.55
Mean: -7.45
Frequency
0
200
400
600
800
−40
−35
−30
−25
−20
−15
−10
−5
0
loudness

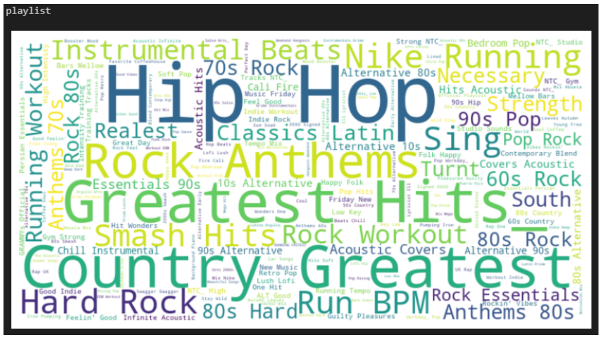

Median: 0.12
Mean: 0.18
Frequency
0
250
500
750
1000
1250
1500
1750
0.0
0.2
0.4
0.6
0.8
1.0
liveness


song_name
Love
feat
Remix
Remastered
Acoustic
Want
Radio
Edit
Heart
Back
Version
Single
Night
Time
Girl
Come
Way
One
Life
Take
Mind
Feel
Day
Know
Song
Dance
Let
Good
Blue
Live
Fire
Boy
Better
Never
Say
Dream
Edit
Now
Man
Light
La
Call
Got
Home
Make


Median: 0.06
Mean: 0.10
Frequency
0
500
1000
1500
2000
2500
0.0
0.2
0.4
0.6
0.8
speechiness


album_names
Best
Love
feat
Remix
Greatest
Hits
Deluxe Edition
Remastered
One
Single
Life
Deluxe
Remixes
Time
Night
Vol
Essential
Motion Picture
Soundtrack
Deluxe Version
EP
World
Song
Back
Music
Light
Day
Girl
Summer
Come
Acoustic
Dream
Let


Median: 4.00
Mean: 3.96
Frequency
0
2500
5000
7500
10000
12500
15000
17500
0
1
2
3
4
5
time_signature


playlist
Hip Hop
Rock Anthems
Greatest Hits
Country Greatest
Instrumental
Beats
Nike
Running
Necessary
Strength
Sing
Classics
Latin
Realest
Smash Hits
Rock Workout
Hard Rock
Run BPM
80s Hard
Rock Essentials
Anthems 80s
Workout
Running
70s Rock
60s Rock
90s Pop
Pop Rock
South
Turnt
80s Rock