**Song Popularity prediction using External Song features and Metadata**

Abhinav Lakhani(0769062)

DAB402 Capstone Project

Prof. Savita Seharawat

Date: 04/04/2022

# 

Table of Contents

[Abstract 3](#_Toc100227245)

[Related Work 4](#_Toc100227246)

[Dataset and Features 5](#_Toc100227247)

[Experimental Design 9](#_Toc100227248)

[Exploratory Data Analysis 10](#_Toc100227249)

[Feature Interaction 13](#_Toc100227250)

[Predictor Feature Interactions 14](#_Toc100227251)

[Categorical feature interactions 16](#_Toc100227252)

[Data Cleaning 19](#_Toc100227253)

[Modeling 21](#_Toc100227254)

[Results 21](#_Toc100227255)

[Conclusion 21](#_Toc100227256)

[Future Work 22](#_Toc100227257)

[References 23](#_Toc100227258)

# Abstract

Record companies invest billions of dollars in new artist talent around the globe each year. In 2017, the music industry generated $8.72 billion in the United States alone (Christman, 2018). Thanks to growing streaming services (Spotify, Apple Music, etc) the industry continues to flourish. Popular songs secure the lion’s share of revenue. The top 10 artists in 2016 generated a combined $362.5 million in revenue (Greenburg, 2017). The question of what makes a song popular has been studied before with varying degrees of success (Pham, Kyauk, & Park). Every song has key characteristics including lyrics, duration, artist information, temp, beat, loudness, chord, etc. Previous studies that considered lyrics to predict a song’s popularity had limited success (Xue & Dupoux). Gaining insight into what actually makes a hit song would provide tremendous benefits for the music industry. In this research we tackle this question by focusing on the hit song classification problem. A database of hit songs according to Spotify song popularity, including basic musical features, as well as more advanced features that capture a temporal aspect. Several different classifiers are used to build and test popularity prediction models. The resulting best model has a good performance when predicting whether a song is a popular versus a lower listed position

Keywords: Clustering, Model Selection, Pipeline, Hyperparameter tuning

Business Questions:

1. Is it possible to predict whether a song would be popular or not only from external characteristics of the song and metadata, such as, Instrumentalness, Danceability, Loudness, Tempo, etc.?
2. Which Factors would help the most in predicting song popularity and which features we can remove in further analysis?
3. How much information does the provided dataset have? i.e. With how much accuracy would we able to predict the popularity of a given song?

For example, If we have less data then what other information would we need to improve the predictions?

Github Link: <github.com/abhinav3398/song-popularity>

# Related Work

In 2011 record companies invested a total of 4.5 billion in new talent worldwide (Investing in music, 2012). Gaining insight into what actually makes a song a hit would provide tremendous benefits for the music industry. This idea is the main drive behind the new research field referred to as “Hit song science” which Pachet (Hit song science, 2012) define as “an emerging field of science that aims at predicting the success of songs before they are released on the market”.

There is a large amount of literature available on song writing techniques (J., 2007; Webb, 1999). Some authors even claim to teach the reader how to write hit songs (Leikin, 2008; Perricone, 2000). Yet very little research has been done on the task of automatic prediction of hit songs or detection of their characteristics.

The increase in the amount of digital music available online combined with the evolution of technology has changed the way in which we listen to music. In order to react to new expectations of listeners who want searchable music collections, automatic playlist suggestions, music recognition systems etc., it is essential to be able to retrieve information from music (M.A. Casey, 2008). This has given rise to the field of Music Information Retrieval (MIR), a multidisciplinary domain concerned with retrieving and analysing multifaceted information from large music databases (Downie, 2003).

Many MIR systems have been developed in recent years and applied to a range of different topics such as automatic classification per genre (Cook., 2002), cultural origin (Smaragdis, 2002), mood (C. Laurier, 2008), composer (D. Herremans, 2003), instrument (S. Essid, 2006), similarity (D. Schnitzer, 2009), etc. An extensive overview is given by (Z. Fu, 2011). Yet, as it appears, the use of MIR systems for hit prediction remains relatively unexplored.

The first exploration into the domain of hit science is due to Dhanaraj and Logan (2005). They used acoustic and lyric-based features to build support vector machines (SVM) and boosting classifiers to distinguish top 1 hits from other songs in various styles. Although acoustic and lyric data was only available for 91 songs, their results seem promising. The study does however not provide details about data gathering, features, applied methods and tuning procedures.

Based on the claim of the unpredictability of cultural markets made by Salganik et al. (2006), Pachet and Roy (2008) examined the validity of this claim on the music market. Based on a dataset they were not able to develop an accurate classification model for low, medium or high popularity based on acoustic and human features. They suggest that the acoustic features they used are not informative enough to be used for aesthetic judgements and suspect that the previously mentioned study (Dhanaraj & Logan, 2005) is based on spurious data or biased experiments.

Borg and Hokkanen (2011) draw similar conclusions as Pachet and Roy (2008). They tried to predict the popularity of music videos based on their YouTube view count by training support vector machines but were not successful.

Another experiment was set up by Ni et al. (2011), who claim to have proven that hit song science is once again a science. They were able to obtain more optimistic results by predicting if a song would reach a top 5 position on the UK top 40 singles chart compared to a top 30-40 position. The shifting perceptron model that they built was based on thus far novel audio features mostly extracted from The Echo Nest1. Though they describe the features they used on their website (Jehan & DesRoches, 2012), the paper is very short and does not disclose a lot of details about the research such as data gathering, preprocessing, detailed description of the technique used or its implementation.

In this research accurate models are built to predict if a song would become hit(popular) or not. For this purpose, a dataset of hits including some unique audio features is compiled. Based on this data different efficient models are built and compared.

# Dataset and Features

This work uses dataset from recent Kaggle competition ("Song Popularity Prediction”, n.d.) which is an exhaustive collection of audio features and metadata for about 50,000 songs. The audio features include attributes about the music track itself, such as song duration, key, audio mode, time signature, The metadata uses more abstract features, such as danceability, energy, instrumentalness, liveness, etc. This Dataset is a subset which was derived from Spotify web api which provides access to user related data, like playlists and music that the user saves in their Music library containing data regarding millions of songs and continuously being updated with new songs and changing user preference (Web API, n.d.).

The dataset used for our analysis has 14 attributes within it, and are described below.

|  |  |  |
| --- | --- | --- |
| Column | data type | description |
| song\_duration\_ms | float64 | The duration of the track in milliseconds. |
| acousticness | float64 | A confidence measure from 0.0 to 1.0 of whether the track is acoustic. |
| danceability | float64 | Describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. |
| energy | float64 | Represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. |
| instrumentalness | float64 | Predicts whether a track contains no vocals. â€œOohâ€ and â€œaahâ€ sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly â€œvocalâ€. |
| key | category | The estimated overall key of the track. Integers map to pitches using standard Pitch Class notation . E.g. 0 = C, 1 = Câ™¯/Dâ™­, 2 = D, and so on. n\_levels: 12 |
| liveness | float64 | Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. |
| loudness | float64 | The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. |
| audio\_mode | category | 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. n\_levels: 2 |
| speechiness | float64 | This 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. |
| tempo | float64 | 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. |
| time\_signature | category | An estimated overall time signature of a track. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure). n\_levels: 4 |
| audio\_valence | float64 | Describes 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). |

Table 1: Data Dictionary

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Attributes** | **count** | **mean** | **std** | **Min** | **25%** | **50%** | **75%** | **max** |
| song\_duration\_ms | 35899 | 193165.8 | 45822.13 | 25658 | 166254.5 | 186660 | 215116 | 491671 |
| acousticness | 36008 | 0.276404 | 0.297928 | -0.01355 | 0.039618 | 0.140532 | 0.482499 | 1.065284 |
| danceability | 35974 | 0.570951 | 0.19001 | 0.043961 | 0.42476 | 0.608234 | 0.718464 | 0.957131 |
| energy | 36025 | 0.683932 | 0.212662 | -0.00168 | 0.539276 | 0.704453 | 0.870503 | 1.039741 |
| instrumentalness | 36015 | 0.036527 | 0.150024 | -0.0044 | 0.000941 | 0.001974 | 0.003225 | 1.075415 |
| liveness | 35914 | 0.198514 | 0.15167 | 0.027843 | 0.111796 | 0.135945 | 0.212842 | 1.065298 |
| loudness | 36043 | -7.4076 | 3.877198 | -32.1179 | -9.57814 | -6.34541 | -4.62071 | -0.87735 |
| speechiness | 40000 | 0.094107 | 0.083591 | 0.015065 | 0.0385 | 0.055881 | 0.118842 | 0.560748 |
| tempo | 40000 | 116.5628 | 26.16791 | 62.05578 | 96.99531 | 113.796 | 128.5174 | 219.1636 |
| audio\_valence | 40000 | 0.580645 | 0.237351 | 0.013398 | 0.398669 | 0.598827 | 0.759635 | 1.022558 |

*Table 2: Numerical Statistics*

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | **key** | **audio\_mode** | **time\_signature** |
| Count | 35935 | 40000 | 40000 |
| Unique | 12 | 2 | 4 |
| top | 0 | 0 | 3 |
| freq | 5175 | 27154 | 23358 |

*Table 3: Categorical Stats*

Our Dataset has 13 Features out of which 10 are numerical(song\_duration\_ms, Acousticness, danceability, energy, instrumentalness, liveness, loudness, speechiness, tempo, audio\_valence) and 3 are categorical(key, audio\_mode, time\_signature). We find that, there are a number of features with values between approximately 0 and 1; such as acousticness, danceability, or liveness among others. Also, all of loudness values are negative.

Chart, bar chart

Description automatically generated Chart, bar chart

Description automatically generated

Figure 1: % of missing values in each feature

The Dataset has 56.8525% of records with missing values but, the density of na values is only 6.26% in the entire dataset.. That is a lot of missing values. If dig deep we find that, loudness, liveness, key, instrumentalness, energy, danceability, acousticness and duration\_ms each has around 10% of records that are missing some information. If we change our perspective from columns to rows, there are about 38% of records with only 1 feature missing in it, 15% records with 2 missing features, 4% records with 3 missing features and <1% records with more than 3 missing features. Maximum number of missing features any record had were 6.

Chart, bar chart

Description automatically generated Chart, bar chart

Description automatically generated Chart, bar chart, histogram

Description automatically generated Chart, bar chart

Description automatically generated

Figure 2: Target and NA values Distribution

If we drill down into each categories and look at the na values distributions we find that there is no imbalance in the categorical features, key, time\_signature, audio\_mode. na values are equally distributed in each levels of categorical features. But when we shift our perspective to only target distribution of the whole dataset, we find that our dataset is imbalanced towards less popular songs having about 63% of observations with low popularity, as was followed in categorical feature levels.

# Experimental Design

Diagram

Description automatically generated

Figure 3: Experimental Design

Because the dataset is imbalanced, we will use stratified cross validation technique, wherein which the folds are made by preserving the distribution of target, in our model selection step. We will also use AUCROC score as our major matric in model training and validation.

As more than half of our records have null values, we tried to predict the popularity of the song only from null values as an experiment to determine if we can gain some information from null values only. First the dataset was converted into boolean values where the true value means that the location in the dataset is null and false meant that the location had some value and is not null. A Logistic Regression model was trained on this dataset which gave a score of 0.4928 (AUCROC), almost same as random prediction(0.5).

As we cannot gain any more information than a random guess from null values, we will try to impute the null values by imputing them using Scikit-Learn Iterative Imputer which models each feature with missing values as a function of other features, and uses that estimate for imputation (Iterative Imputer, n.d.). It does so in an iterated round-robin fashion: at each step, a feature column is designated as output and the other feature columns are treated as inputs . A regressor is fit on for known . Then, the regressor is used to predict the missing values of . This is done for each feature in an iterative fashion, and then is repeated for max\_iter imputation rounds. The results of the final imputation round are returned. After NA values imputations, the dependent variables are fit into a normal distributions to impute the outliers. Then, the remaining few outliers are removed completely, as they cannot be imputed further, using Local Outlier Factor which measures the local deviation of the density of a given sample with respect to its neighbors. It is local in that the anomaly score depends on how isolated the object is with respect to the surrounding neighborhood. More precisely, locality is given by k-nearest neighbors, whose distance is used to estimate the local density. By comparing the local density of a sample to the local densities of its neighbors, one can identify samples that have a substantially lower density than their neighbors. These are considered outliers (LocalOutlierFactor, n.d.).

After cleaning the data and getting insights from Exploratory Data Analysis, we will build and fit various Classifiers using 5 fold stratified cross validation, while also tuning their hyperparameters which maximizes the AUCROC score to find the top models which best generalize the dataset. Following model selection, in model validation we will look at the performance of selected models and determine whether the model score is sufficient enough to make a pipeline which can be used in production or do we need further data.

Chart, line chart

Description automatically generated

Figure 4: ROC Curve of missing value indicators as features

## Exploratory Data Analysis

Looking at the distributions of each feature, we observe that song\_duration had the distribution that is closest to normal distribution out of all the features, as can be seen from qq plot which is closest to the solid line. Acousticness, liveness and speechiness had right skewed distribution. On the other hand, loudness and energy are left skewed with bimodal distributions. Danceability, audio\_valance and tempo showed multimodal distributions with later slightly skewed towards right side. Instrumentalness had slightly weird distribution, which will be investigated further.

Diagram

Description automatically generated with low confidence

Figure 5: Feature Distributions

In the categorical feature distributions, key had almost normal distribution except key 11, which was not being used in most of the songs. Audio mode had high imbalance towards value 0. Similarly, time\_signature was also imbalanced towards value 3 and 4.

Finally, from qq plots we observe that, acousticness, danceability, energy, instrumentalness, liveness, speechieness and audio\_valence has an “S” shaped distribution resembling closer to the classic sigmoid distribution. Hence, we can apply an inverse sigmoid trans formation on these variables to fit to normal distribution which would help in model building step later. After which we can apply power transformations (Atkinson, 2020) on all of the numerical variables to further fit to Gaussian probability distribution.

Sigmoid Function:

Inverse Sigmoid:

Chart

Description automatically generated

Figure 6: qq plots against standard normal distribution as shown by solid line

### Feature Interaction

After learning more about each individual feature, we now want to see them interacting with one another. We will start with studying the impact of the target song\_popularity on each individual feature, then look at correlations and relationships between the predictor features, before moving on to even higher dimensional relationships.

We have seen all the feature distributions, now we want to investigate whether they look different based on the target value.

Chart

Description automatically generatedFigure 7: Target Impact on independent numerical variables.

There are no features that show strong differences in their distributions for popular vs unpopular songs. No smoking guns here. Some features like energy, audio\_valence, and perhaps tempo show some degree of difference between target classes. Others, like song\_duration\_ms or liveness appear to overlap almost perfectly.

Chart, bar chart

Description automatically generated

Figure 8: Target Impact on independent categorical variables.

This kind of view does tell us as much, though, because the imbalance between popular and unpopular songs appears to be pretty universal for all of the features. In order to see the relative percentages of song\_popularity = 1 or 0 (or true vs false as we're expressing it here) we need to compute those fractions for every feature & value combination.

### Predictor Feature Interactions

Looking at the correlation matrix, we find that, there iss a strong anti-correlation between acousticness vs energy and loudness, respectively. Consequently, energy and loudness share a strong correlation. Also, none of the features individually show a notable correlation with the target song\_popularity.

Chart, scatter chart

Description automatically generated

Figure : Correlation Matrix

### Categorical feature interactions

Chart, surface chart

Description automatically generated

Figure : Interaction between energy and time\_signature

There are clear differences between the energy peaks for time\_signature = 2 vs 4. Smaller differences exist between the other values. 3 and 4 are the categories with the large numbers, and 4 is clearly more energetic than 3.

Chart

Description automatically generated

Figure : Interaction between time signature and key according to energy

For time\_signatures 2 and 5 we have no instances of key == 11. This is no big surprise, since those three values are already rare individually, which makes their combinations even more rare. The lower keys and time signatures have less energy, on average, than the higher ones. In particular key 9 or 11.

Chart, scatter chart

Description automatically generated

Figure : relation between Energy and Loudness

The relationship looks pretty solid; not just built on outliers or strange. The 2 peaks of energy don't map 1:1 to the 2 peaks of loudness (which has a relatively similar distribution shape).

Chart

Description automatically generated

Figure : Correlation between energy, loudness and time signature

Other than the obvious sparsity of the samples with key 2 or 5, we see slightly different relations between the better populate 3 and 4 facets. The key == 4 sample appears to have on average fewer values below -15

Chart

Description automatically generated

Figure : Exploratory analysis of Instrumentalness

This is a useful result, because it points towards a relatively well behaved part of the distribution below 0.01. We know that there are other values above that threshold of 0.01, so let's count both groups:

|  |  |  |  |
| --- | --- | --- | --- |
| **index** | **subset** | **n** | **percentage** |
| TRUE | main\_subset | 32763 | 0.90970429 |
| FALSE | side\_subset | 3252 | 0.09029571 |

Figure : Distribution of the subset of instrumentalness

About 91% of instrumentalness has values below 0.01 and fall into the "main distribution". The remaining 9% are above that threshold and we will call that the "side distribution".

### Data Cleaning

#### NA value Imputation

Chart, histogram

Description automatically generated

Figure : Distribution of Simple mean value imputer, KNN imputer and Iterative Imputer

Simple Imputer simply imputes the outlier value by the mean value of the distribution as can be seen by the spike in the centre of the distribution. KNN imputer uses Imputation for completing missing values using k-Nearest Neighbors. Iterative Imputer is a multivariate imputer that estimates each feature from all the others. A strategy for imputing missing values by modeling each feature with missing values as a function of other features in a round-robin fashion. Uses by default the BayesianRidge model to impute.

#### Outlier Removal

By transforming acousticness, danceability, energy, instrumentalness, liveness, speechieness and audio\_valence using inverse sigmoid transformation and later applying power transformation to all of the numerical variables, we see that those variables now have more normal distributions as compared to before, and the outliers at the tails of the distribution are now imputed to be inside the normal distribution, as can be seen from the qq plot.

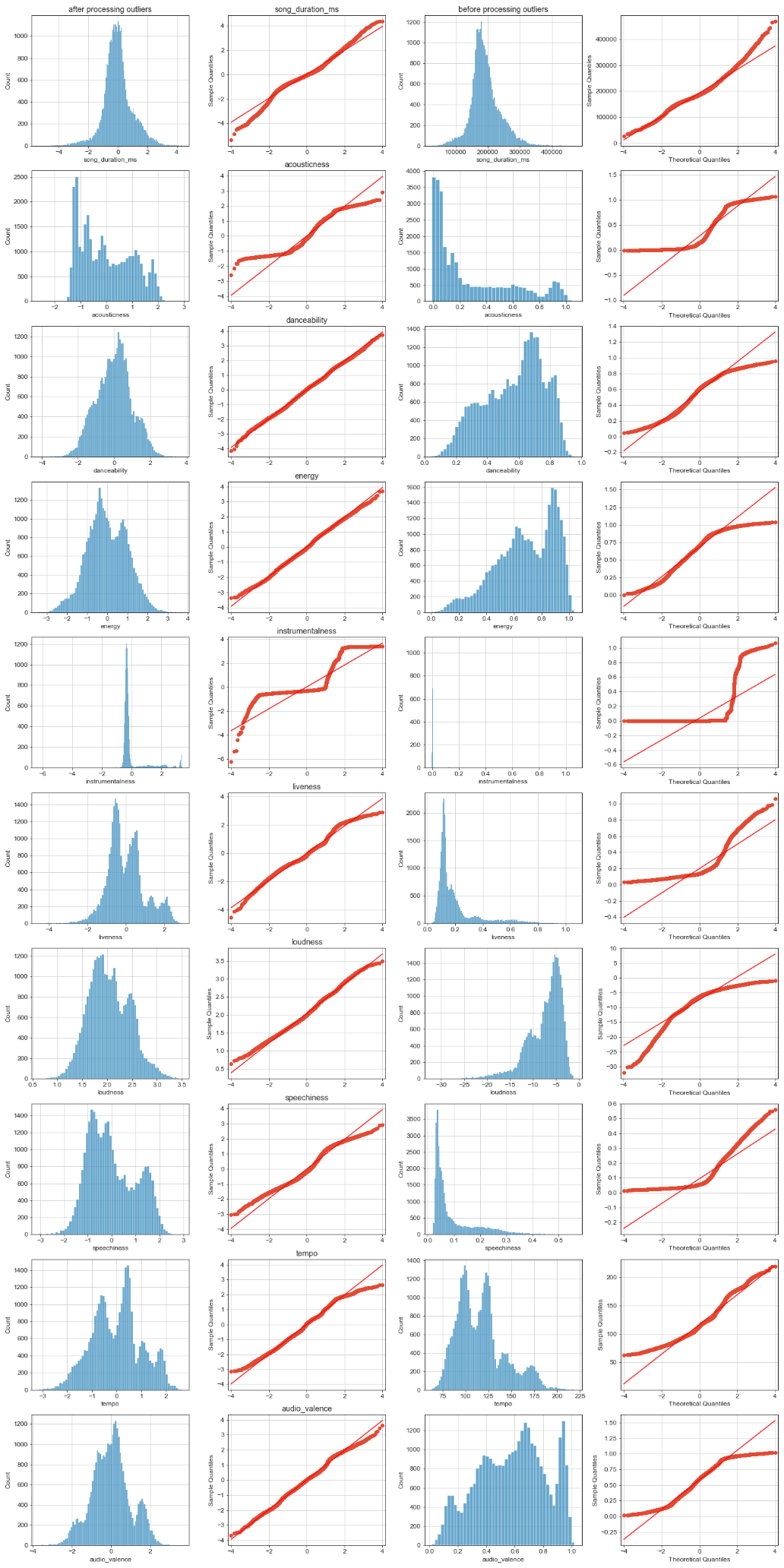


Figure : Comparisons of numerical distributions before outlier removal(plots on right side) and after outlier removal by transforming the target distribution to normal distribution.

## Modeling

Before modeling we split our dataset into test and train, with test set being 20% of the entire dataset. After that we apply all of the data transformation steps we performed in Exploratory Data Analysis. Thereafter, we train various scikit-learn classifiers and tune their hyperparameters to find the top 15 models with highest AUCROC score.

## Results

Chart, bar chart

Description automatically generated

Figure : Model Scores on Test set.

After getting the top model we evaluate those models on test set and obtain the score. We see that the best models had score less than 60%. If we further use a expand our search for the best model beyond scikit-learn classifiers using TPOT library (Olson, n.d.), we find that XGBClassifier, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning model, was able to get a sloghtly higher score of 66%.

# Conclusion

Looking at the low model performance, we conclude that we need more information about the song being classified. As we have used most of the external information about the song, in our modeling and still got unsatisfied score, we will need internal characteristics of the song such as song lyrics or the actual audio data or at least portion of the song being predicted.

## Future Work

Here we only looked at machine learning algorithms. But we may look at more complex models available in deep learning libraries such as neural nets, auto-encoders, transformers etc. to build a more complex model for prediction. Or we could find more external characteristics which ware not present in this literature.

# References

*"Song Popularity Prediction”*. (n.d.). From Kaggle: https://www.kaggle.com/c/song-popularity-prediction

Atkinson, A. C. (2020). The Box-Cox Transformation: Review and Extensions. London: The London School of Economics. From http://eprints.lse.ac.uk/103537/1/StatSciV4.pdf

Borg, N., & Hokkanen, G. (2011). What makes for a hit pop song? What makes for a popular song? From http://cs229.stanford.edu/proj2011/BorgHokkanen-WhatMakesForAHitPopSong.pdf

C. Laurier, J. G. (2008). Multimodal music mood classification using audio and lyrics. In *Machine Learning and Applications, 2008 ICMLA’08. Seventh International Conference* (pp. 688–693). IEEE.

Christman, E. (2018, 3 22). *U.S. Music Industry Hits Highest Revenue Mark in a Decade, Fueled by Paid Subscriptions*. From Billboard: https://www.billboard.com/articles/business/8257558/us-music-industry-2017-highest-revenue-in-decade-fueled-paid-subscriptions

Cook., G. T. (2002). Musical genre classification of audio signals. In *Speech and Audio Processing* (pp. 10(5):293–302). IEEE transactions.

D. Herremans, K. S. (2003). *Classification and generation.* Working paper - University of Antwerp.

D. Schnitzer, A. F. (2009). A filter-and-refine indexing method for fast similarity search in millions of music tracks. In *Proceedings of the 10th International Conference on Music Information Retrieval.* ISMIR09.

Dhanaraj, R., & Logan, B. (2005). Automatic prediction of hit songs. In *Proceedings of the International Conference on Music Information Retrieval,* (pp. 488–91).

Downie, J. (2003). Music information retrieval. *Annual review of information science and technology*, 37(1):295–340.

Greenburg, J. O. (2017, 12 6). *The World's Highest-Paid Musicians Of 2017*. From Forbes: https://www.forbes.com/sites/zackomalleygreenburg/2017/12/06/the-worlds-highest-paid-musicians-of-2017/?sh=2544bbae530e

(2012). *Investing in music.* IFPI (International Federation of). From http://www.ifpi.org/content/library/investing\_in\_music.pdf

*Iterative Imputer*. (n.d.). From Scikit-Learn: https://scikit-learn.org/stable/modules/generated/sklearn.impute.IterativeImputer.html

J., B. (2007). *Craft and Business of Songwriting.* F & W Publications.

Jehan, T., & DesRoches, D. (2012). *EchoNest Analyzer Documentation.* From http://developer.echonest.com/docs/v4/\_static/AnalyzeDocumentation.pdf

Leikin, M. (2008). *How to Write a Hit Song, 5th Edition.* Hal Leonard.

*LocalOutlierFactor*. (n.d.). From Scikit-Learn: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html

Logan, R. D. (2005). Automatic prediction of hit songs. In P. o. Retrieval.

M.A. Casey, R. V. (2008). Content-based music information retrieval: Current directions and future Challenges. *Proceedings of the IEEE*, 96(4):668–696.

M.J. Salganik, P. D. (2006). Experimental study of inequality and unpredictability in an artificial cultural market. In *science, 311(5762)* (pp. 854–856,).

Olson, R. S. (n.d.). *TPOT*. From epistasislab: http://epistasislab.github.io/tpot/

Pachet, F. (2012). Hit song science. In *Tzanetakis & Ogihara Tao, editor, Music Data* (pp. 305–326).

Pachet, F., & Roy, P. (2008). Hit song science is not yet a science. In *The 9th International Conference on Music Information Retrieval* (pp. 355–360). ISMIR.

Perricone, J. (2000). *Melody in Songwriting: Tools and Techniques for Writing Hit Songs.* Berklee Press.

Pham, J., Kyauk, E., & Park, E. (n.d.). *Predicting Song Popularity.* Stanford University, Computer Science. Stanford University. From http://cs229.stanford.edu/proj2015/140\_report.pdf

Roy, F. P. (2008). Hit song science is not yet a science. In *he 9th International Conference on Music Information Retrieval* (pp. 355–360). (ISMIR.

S. Essid, G. R. (2006). Musical instrument recognition by pairwise classification strategies. In *Audio, Speech, and Language Processing* (pp. 14(4):1401–1412). IEEE Transactions.

Smaragdis, B. W. (2002). Combining musical and cultural features for intelligent style detection. ISMIR.

Watts, D. J., J., M., Dodds, P. S., & Salganik. (2006). Experimental study of inequality and unpredictability in an artificial cultural market. *science*, 854-856.

*Web API*. (n.d.). From Spotify for Developers: https://developer.spotify.com/documentation/web-api/)

Webb, J. (1999). *Tunesmith: Inside the Art of Songwriting.* Hyperion.

Xue, A., & Dupoux, N. (n.d.). *Predicting A Song’s Commercial Success Based on Lyrics and Other Metrics.* From http://cs229.stanford.edu/proj2014/Angela%20Xue,%20Nick%20Dupoux,%20Predicting%20the%20Commercial%20Success%20of%20Songs%20Based%20on%20Lyrics%20and%20Other%20Metrics.pdf

Y. Ni, R. S.-R. (2011). *Hit song science once again a science?*

Z. Fu, G. L. (2011). A survey of audio-based music classification and annotation. Multimedia, IEEE Transactions.

Pachet, Franois, and Pierre Roy. ”Hit Song Science Is Not Yet a Science.” ISMIR. 2008.

Ni, Yizhao, et al. ”Hit song science once again a science.” 4th International Workshop on Machine Learning and Music, Spain. 2011.