Journal on Interactive Systems, 2023, 14:1, doi: 10.5753/jis.2023.3137

Collaboration-Aware Hit Song Prediction

Mariana O. Silva [**Universidade Federal de Minas Gerais** | ***mariana.santos@dcc.ufmg.br***]
Gabriel P. Oliveira [**Universidade Federal de Minas Gerais** | ***gabrielpoliveira@dcc.ufmg.br***]
Danilo B. Seufitelli [**Universidade Federal de Minas Gerais** | ***daniloboechat@dcc.ufmg.br***]
Mirella M. Moro [**Universidade Federal de Minas Gerais** | ***mirella@dcc.ufmg.br***]

Abstract In a streaming-oriented era, predicting which songs will be successful is a significant challenge for the music industry. Indeed, there are many efforts in determining the driving factors that contribute to a song's success, and one potential solution could be incorporating artistic collaborations, as it allows for a wider audience reach. Therefore, we propose a multi-perspective approach that includes collaboration between artists as a factor for hit song prediction. Specifically, by combining online data from Billboard and Spotify, we tackle the problem as both classification and *hit song placement* tasks, applying five different model variants. Our results show that relying only on music-related features is not enough, whereas models that also consider collaboration features produce better results.

Keywords: *Hit Song Science, Hit Song Prediction, Music Information Retrieval, Music Data Mining, Machine Learning*

1 Introduction

Predicting hit songs is a major open issue for the music industry, as such prediction allows it to improve its revenues by focusing on potential hits or even perfecting the characteristics of a song so that it becomes popular and commercially well. Nevertheless, this is not a novel problem, as there are many efforts to find the driving factors that shape the success of songs, which have achieved quite a critical mass and are now part *Hit Song Science* (HSS) (Pachet, 2011), defined as "an emerging field of science that aims at predicting the success of songs before they are released on the market".

Most HSS approaches employ a classification algorithm (Araujo et al., 2017, 2019; Cosimato et al., 2019; Dhanaraj and Logan, 2005; Interiano et al., 2018; Ni et al., 2011), a regression model (Kim et al., 2014; Nunes and Ordanini, 2014; Zangerle et al., 2019), or both (Kim et al., 2014; Martín-Gutiérrez et al., 2020). All techniques work over different song features, which may be divided into *intrinsic*, such as acoustic or lyric-based data (Dhanaraj and Logan, 2005; Nunes and Ordanini, 2014; Vötter et al., 2021); or *extrinsic*, such as social media buzz (Araujo et al., 2017; Cosimato et al., 2019; Kim et al., 2014). Few consider both types of features together (Interiano et al., 2018).

Given the inherent limitations in the availability and collection of music data, it is naturally more complex to consider all possible extrinsic factors than intrinsic factors when predicting the success of a song. Indeed, although some extrinsic factors have been considered in previous works, an unexplored feature is artistic collaboration. Collaboration between artists is an increasingly common practice, as it allows the songs to reach wider audiences (Silva and Moro, 2019). Figure 1 shows a boom in collaboration from the mid-1990s on Billboard Hot 100.

Even with such compelling information, we are the first to investigate collaboration between artists and its potential influence on the performance of hit song prediction. This work is an extension of a previously published paper (Silva et al., 2022), motivated by two key research questions (**RQs**):

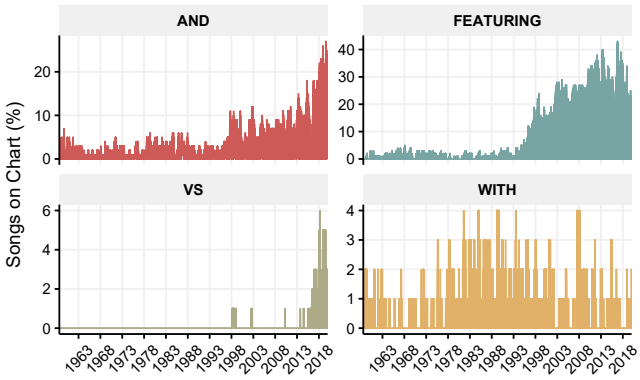


Figure 1. Pairwise historical frequency of collaboration types on Billboard Hot 100 Chart (1958 - 2020). Collaboration types include: *and* (usually a set of artists with equal rights), *featuring* (usually a short appearance), *vs.* (usually a DJ contest) and *with* (usually a duet). Note distinct y-axis scales.

(RQ1) *How much does considering artists' collaborations affect hit song prediction?* and **(RQ2)** *Considering song, album and artist perspectives, how does each of them interfere in the prediction model outcome?*

To answer both questions, we assess the hit song prediction (HSP) problem from two different tasks: binary classification and *hit song placement* (Section 3.1). Both tasks work over a multi-perspective approach based on song, album and artist features (Section 3.2). We then consider five HSP variant models that learn from different feature combinations, including only song perspective, all three perspectives, only collaboration-based features, song perspective combined with collaboration, and all three perspectives plus *collaboration between artists*, which is a novel approach proposed in this study (Section 3.3). Finally, we evaluate the performance of our proposed representations by using specific experimental setups and metrics (Section 5.1) that evaluate the prediction results (Section 5.2) and interpret the learned models to identify the most important features (Section 5.3). Overall, our contributions are summarized as follows.

1. To the best of our knowledge, we are the first to explore artists' collaboration as a predictor in hit song prediction;
2. We also are the first to define the Hit Song Prediction prob-

Table 1. Summary of Related Work

Ref† / Information	Internal features	External features	Album features	Artist features	Classification	Regression	Spotify data	Billboard data	Dataset size‡	Collaboration
(Dhanaraj and Logan, 2005)	x				x				1.7k	
(Pachet and Roy, 2008)	x	x			x				33k	
(Bischoff et al., 2009)		x		x	x				317k	
(Ni et al., 2011)	x				x				6k	
(Kim et al., 2014)		x		x	x	x		x	178	
(Nunes and Ordanini, 2014)	x			x	x			x	2.4k	
(Ren et al., 2016)	x	x		x	x				2k	
(Araujo et al., 2017)	x	x	x	x		x			194	
(Yang et al., 2017)		x				x			20k	
(Interiano et al., 2018)	x	x		x	x				500k	
(Araujo et al., 2019)	x				x		x		16.4k	
(Cosimato et al., 2019)		x			x		x	x	N/A	
(Zangerle et al., 2019)	x					x		x	1M	
(Martín-Gutiérrez et al., 2020)	x	x		x	x	x	x		102k	
(Kim and Oh, 2021)	x				x		x	x	6.2k	
(Vötter et al., 2021)	x	x			x	x		x	81.2k	
(Silva et al., 2022)	**x**	**x**	**x**	**x**	**x**		**x**	**x**	**911k**	**x**

† sorted by year; ‡ number of songs; x = yes; k = thousand; M = million; N/A = not available

lem as a *placement* task;

3. The performance and feature importance results show relying only on song-related features is not enough, whereas models that also consider collaboration features produce better results;
4. Moreover, the results show that features extracted from the *artist* perspective, mainly collaborative information, are the most significant predictors for songs popularity.

2 Related Work

Music and Computer Science have a long, fruitful relationship (Roads, 1996; Schedel and Young, 2005; Costa et al., 2020). Indeed, computer systems and computing techniques have helped the Music community (and, sometimes, the Arts community as well) with a myriad of different problems, from analyzing Antonio Carlos Jobim's songs (Almada et al., 2019) to generating music playlists (de Almeida et al., 2017) and enhancing the quality of degraded music (Serra et al., 2021). Nonetheless, this work focuses on a specific task that may directly impact the music industry: understanding how a song may become a hit. In such a context, the area known as Hit Song Science (HSS) aims to understand the success of a song as a product of its technical and acoustic features, such as timbre, duration, tone, and energy, among others (Dhanaraj and Logan, 2005).

Each song may be represented by a set of features, such as loudness (our perception of sound amplitude or volume), pitch (the song's harmonic content, including chords and melody), and timbre (aka. the tone quality) (Serrà et al., 2012). A common grouping of musical features separates them into internal and external (also referred as intrinsic/extrinsic) (Yang et al., 2017). *Internal features* relate directly to the content of each song, including different aspects of audio properties, song lyrics, and its artists (Araujo et al., 2019; Dhanaraj and Logan, 2005; Ni et al., 2011; Nunes and Ordanini, 2014; Vötter et al., 2021; Zangerle et al., 2019; Kim and Oh, 2021). For example, Dhanaraj and Logan (2005) use support vector machine based on features computed from audio Mel-frequency cepstral coefficients (MFCC) and song lyrics to decide if a song will appear in music charts; Ni et al. (2011) show how tempo, duration, loudness and harmonic simplicity correlate with the evolution of musical trends; whereas Pachet and Roy (2008) use many audio features (e.g., style, genre, tempo, mood, language, rhythm) to claim that hit song science is not yet a science, when using such information in classification algorithms.

Then, *external features* relate to information that cannot be derived from the song itself, which includes social events, marketing and album cover design. For example, Salganik et al. (2006) create an artificial music market and study the relevance of social influences (i.e., other listeners' behavior) on finding hit songs. The first half of Table 1 summarizes and compares the types of features used in such works. Note that, back in 2008, Pachet and Roy (2008) argued that acoustic characteristics were not yet sufficient to predict popularity. Hence, some studies consider the impact of social factors and other external features on music's success (Bischoff et al., 2009; Ren et al., 2016; Araujo et al., 2017; Cosimato et al., 2019). Another variable is to consider the performance of a song or a whole album. For instance, Cosimato et al. (2019) explore the relationship between *album* popularity and the buzz about it on social media, whereas Araujo et al. (2017) relate *album* sales and the sentiments about it on tweets. However, there is still plenty of room to explore artists' collaboration as an influential social aspect, as we do.

The only work that considers a kind of collaboration is by Calefato et al. (2018). Nonetheless, a collaboration of artists is defined very differently from the standard definition of collaboration, where people act together toward a common goal (Martins et al., 2021). Calefato et al. (2018) referred to artist collaboration as *overdubbing*, i.e., one new track is mixed with an existing audio recording (e.g., voice over an instrumental track). The dataset considers song and author features extracted from an artistic collaboration network (Songtree). Then, success metrics include the number of song followers, times played, and so on. Note that there is no evaluation regarding formal music rankings, no inclusion of any internal features of the songs besides those extracted from the social network, and the closed network features also give success.

As summarized in the second half of Table 1, most approaches model the hit song prediction as classification (Araujo et al., 2019; Cosimato et al., 2019; Dhanaraj and Logan, 2005; Interiano et al., 2018; Kim and Oh, 2021; Ni et al., 2011) or regression (Araujo et al., 2017; Yang et al., 2017; Zangerle et al., 2019) tasks, with a few exceptions that model the problem as both tasks (Kim et al., 2014; Martín-Gutiérrez et al., 2020; Vötter et al., 2021). Among the three approaches, the classification may be the most straightforward way to separate hits from non-hits as its output can be exactly binary. Still, regression models were explored, because of their advantage of returning a continuous outcome based on the value of one or multiple variables (Kim et al., 2014; Martín-Gutiérrez et al., 2020; Nunes and Ordanini, 2014). For example, Nunes and Ordanini (2014) relate the type of instruments and the number of instruments audible to a pop song's popularity in the Billboard Hot 100 chart.

Another important variable when predicting hit songs is to define what a hit is, for which there is no consensus as a hit may belong: Spotify's Top 50 Global ranking (Araujo et al., 2019), Billboard Top 200 Albums (Cosimato et al., 2019), number 1 song in the country (Dhanaraj and Logan, 2005), UK Top 100 Singles Chart (Interiano et al., 2018) or its top five (Ni et al., 2011), Billboard Hot 100 (Kim et al., 2014; Zangerle et al., 2019) or its top 10 (Kim and Oh, 2021) or number one (Nunes and Ordanini, 2014), or the most popular songs on a Taiwanese social network (Yang et al., 2017).

Finally, which dataset to use is also critical for hit song prediction. Few works in our comparison study (Table 1) actually use data available on the Web, such as Billboard charts (Kim et al., 2014; Nunes and Ordanini, 2014; Zangerle et al., 2019), Spotify features (Araujo et al., 2019; Martín-Gutiérrez et al., 2020) or both (Cosimato et al., 2019; Kim and Oh, 2021). da Silva et al. (2020) introduced a dataset with lyrics, genre annotations, metadata, and audio features extracted from the Vagalume[1] Brazilian platform. The dataset contains 96,458 songs (in English, Spanish, Portuguese and Brazilian Portuguese) from 15,310 artists who have appeared at least once in Vagalume's Top ranking. Nonetheless, there is no information on how or when such ranking is defined, which is critical for any classification method.

To get a more extensive set of hit songs and their data, we have created MusicOSet (Silva et al., 2019a,c), an open dataset of musical elements suitable for music data mining that contains 57 years (1962-2019) on the Billboard Hot 100 charts. We now merge it with data from Spotify to get data about the streaming era. In total, this dataset includes over 911 thousand songs (details in Section 4.1).

Overall, there are many critical differences between our research and the aforementioned related work that standout and advance such a state of the art: (*i*) we consider both intrinsic and extrinsic features of 911 thousand songs; (*ii*) besides such a song perspective, we also include data from albums and artists in the prediction model; (*iii*) we are the first to expand the artists' features to include their collaborations, which is a key factor in recent hits; (*iv*) we evaluate the hit song prediction problem using two different tasks (i.e., binary classification and *hit song placement*) and five variant models – based on acoustic features only (the most common configuration in the related work), on song-album-artists perspectives alone and with collaboration among artists as well; and (*v*) we *interpret* the predictions by evaluating and explaining the contribution of each feature to the results.

[1] Vagalume: `https://www.vagalume.com.br/`

3 Hit Song Prediction

We now detail our multi-perspective approach to tackle the Hit Song Prediction (HSP) problem as defined in Section 3.1. Then, Section 3.2 covers the multi-perspective features used as hit song predictors, whereas Section 3.3 describes the designed HSP models to assess the different musical features.

3.1 Problem Definition

To extend our previous work (Silva et al., 2022), we assess the Hit Song Prediction problem using two different tasks: binary classification and *hit song placement*, summarized as follows.

Binary Classification. Given a song, the task is to predict whether it will be a hit or not. Formally, let $\mathcal{X}$ represent a set of songs sorted by their release date, and $\mathcal{Y} = \{1, 0\}$ be the label space, where 1 indicates a hit song and 0 indicates a non-hit song. The objective of binary classification is to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ using the training set $\{(x_i, y_i) \mid 1 \leq i \leq m\}$, where $x_i \in \mathcal{X}$ represents the features of a song, and $y_i \in \mathcal{Y}$ denotes the corresponding target value. It is crucial to perform the train-test split at a specific time t because the chronological order of the songs is crucial for the prediction.

Hit Song Placement. Given a set of hit songs $\mathcal{H} = \{s_1, s_2, \cdots, s_n\}$ sorted by a popularity measure, the Hit Song Placement task aims to predict the rank $r_{s'}$ of a new song s' correctly placed within the ranked list of hit songs. The predicted rank $r_{s'}$ should satisfy the following condition: $r_{s_i} \geq r_{s'} \geq r_{s_j}$ where $(s_i, s_j) \in \mathcal{H}$ are the two hit songs immediately above and below s' in the ranking, respectively. The goal of the task is to accurately predict the rank of the new song s', which provides information on its overall success and its potential level of success on the charts. Unlike traditional regression tasks, the Hit Song Placement task involves predicting the ordinal position of a song in a ranking rather than a continuous numerical value.

Suppose we have a set of hit songs $\mathcal{H} = \{s_1, s_2, s_3, s_4, s_5\}$ sorted by their popularity measure, where $rank(s_1) < rank(s_2) < rank(s_3) < rank(s_4) < rank(s_5)$. A new song s' will be released, and we want to predict its placement in the top chart. To do so, we need to find the correct position for s' between two existing hit songs in the ranking. For example, if our prediction algorithm outputs that s' should place between s_3 and s_4, then the predicted placement for s' is 4 (i.e., it will be the fourth song in the ranking). Note that the placement task is different from the common regression task, which only predicts a numerical value of a measure of success (e.g., the number of streams, sales, etc.). In the placement task, we aim to predict the song's position in the ranking, enabling us to determine its overall success and success level.

Hit song prediction also requires defining what a *hit* is. Along similar lines of previous research (Zangerle et al., 2019; Kim et al., 2014), we define *hits* as songs that appear on the weekly Billboard Hot 100 chart at least once. Such a definition is not a consensus over related work (Section 2), but it is a significant one, as there are thousands of songs playing everywhere and making it to the Hot 100 (even at the 99^{th} position) is a sign of success.

3.2 Multi-Perspective Features

The success and popularity of a song may be affected by numerous features, including those internal/intrinsic and external/extrinsic to a song (see Section 2). From a multi-perspective approach, we explore features within three musical context factors: (*i*) *song*, which includes acoustic characteristics as well as release information; (*ii*) *album*, which includes album type and the number of tracks; and (*iii*) *artist*, which captures artists' collaboration profile quantitatively and qualitatively, as well as the number of genres and albums the artist owns. Table 2 briefly defines each perspective and its features, which are also discussed next.

Song Perspective. We consider song-based features of two types: *intrinsic* and *extrinsic*. As *intrinsic* factors, we consider a set of high-level acoustic fingerprints provided by Spotify API[2] that professionals and researchers can easily interpret. For the latter, we consider the song track number, number of collaborating artists, number of countries in which this song is available on Spotify, and number of years a song has been on the Hot 100 charts.

Album Perspective. In the Spotify API, metadata referring to song albums is also available. Here, we consider two of them: the album type (*album_type*), which classifies albums as *album*, *single* or *compilation*); and the total number of tracks the album has (*album_total_tracks*).

Artist Perspective. We also use Spotify's API to collect the total number of artists' genres and albums, respectively, as proxies for style diversity and artist productivity. In addition to these two factors, based on previous studies showing that artist connections can affect musical success (Silva et al., 2019b; Silva and Moro, 2019), we also consider metrics that capture collaboration between artists. Specifically, based on the ideas from (Silva et al., 2019b), we build a collaboration network where weighted edges connect artists who have collaborated on one song (or more). Next, we compute seven topological metrics and apply a clustering algorithm (K-Means), which results in three well-defined artists' communities with distinct collaboration patterns: *Diverse* for highly collaborative and influential artists; *Regular* for typically collaborative artists; and *Absent* for bands and non-collaborative artists. Then, our Artist perspective considers not only the seven topological metrics but also the three collaboration profiles identified. Figure 2 depicts an overview of the collaboration-based feature creation process.

[2]Spotify API: `https://developer.spotify.com`

3.3 HSP Variant Models

As our main contribution is to explore distinct factors from different perspectives in hit song prediction, we compare variants of the hit song prediction (HSP) model concerning the effect of various feature combinations, including (*i*) relying only on song-based features; (*ii*) including different perspectives (i.e., album and artists); (*iii*) relying only on collaboration-based features; (*iv*) combining song- and collaboration-based features; and (*v*) considering all features. Thus, the following variants are designed for comparison.

HSP-song. A variant of the HSP model considering only information based on the song perspective (which is a common solution from related work).

HSP-three. A variant of the HSP model considering information based on the three perspectives (song, album, artist), without considering collaboration-based factors.

HSP-collab. A variant of the HSP model considering only information based on collaboration-based features.

HSP-song-collab. A variant of HSP-song model considering collaboration-based features.

HSP-three-collab. A variant of HSP-three model considering collaboration-based features.

4 Implementation

This section describes the implementation setup followed for all HSP variant models. In Section 4.1, we describe the preprocessing steps applied to the input data of the models. Then, Sections 4.2 and 4.3 list the learning algorithms used in each HSP variant model.

4.1 Data and Preprocessing

To predict hit songs, we use MusicOSet (Silva et al., 2019a,c), an open dataset of musical elements suitable for music data mining that contains 57 years (1962-2019) of the Billboard Hot 100 charts. Such data is integrated with Spotify's additional information, including musical metadata and acoustic fingerprint features. Such acoustic fingerprints are condensed digital summaries of a song's phonic features that capture the music style and creative experience (Ren et al., 2010). In addition, the dataset also provides popularity information at three levels: song, album and artist.

We focus our analyses on songs from 1995 to 2019 to avoid potential bias from three scenarios. First, there are distinct sale patterns from the LP-CD-dominated market before 1995 and the potential COVID impact starting in 2020. Second, as already pointed out by Interiano et al. (2018); Zangerle et al. (2019), music preferences evolve as each decade has different music genre trends, styles and movements, e.g., grunge in the early nineties. Notice that such changes may also add noise to any analyses or predictions. For example, trying to predict a 2022 hit song based on data from many decades before is certainly a recipe for failure. Third, we also reduce possible bias resulting from changes in the phonographic sector due to technological innovations (e.g., easier distribution and commercialization) and pandemic isolation policies.

Table 2. Description of considered features, separated by perspective.

	Name	Description
Song Features ♫		
Intrinsic	Key	the estimated overall key of a song, mapped as an integer number (e.g. $C = 0, C\# = 1$)
	Loudness	general loudness measured in decibels (dB)
	Mode	general modality of a song (i.e., major or minor)
	Time Signature	amount of beats in each bar (measure)
	Tempo	song speed measured in beats per minute (BPM)
	Acousticness	probability of a song to be acoustic or not
	Danceability	suitable for dancing (a combination of tempo, rhythm, etc.)
	Energy	intensity and activity of a song by combining dynamic range, perceived loudness, timbre, onset rate, general entropy, etc.
	Instrumentalness	probability of a song to be instrumental, i.e., without vocals
	Liveness	presence of an audience in a song (higher liveness value means higher probability of being performed live)
	Speechiness	probability of a given song to have spoken words
	Valence	positiveness within a song (high values mean happier songs)
	Duration (ms)	duration of a song in milliseconds
	Explicit	if the lyrics of a song have explicit content
Extrinsic	Available Markets	number of countries in which the song is available on Spotify
	Number of Artists	number of artists who sing that song
	Track Number	track number in the album
	Years on Hot 100	number of years a song has been on Billboard Hot 100 charts
Album Features ■		
	Album Total Tracks	album total number of songs
	Album Type	whether it is a single, a full album or a compilation
Artist Features ♦		
Base	Number of Albums	number of albums released by the artist
	Number of Genres	number of genres of the artist
Collaboration	Betweennness	a centrality index for how many times a given node appears in the shortest path of the other nodes in the network
	Closeness Centrality	average value of the shortest paths between the given node and all of the other nodes in the network
	Clustering Coefficient	probability of two neighbors of a given node also being neighbors
	Cluster	artist cluster: 1 (*Absent*), 2 (*Diverse*) or 3 (*Regular*)
	Collaboration Profile	artist level of collaboration: *Absent*, *Diverse* or *Regular*
	Eccentricity	the longest shortest path between a given node and all other nodes in the network
	Eigenvector Centrality	how influential a given node is within the network
	Weighted Degree	number of edges incident on it multiplied by the weights of the edges

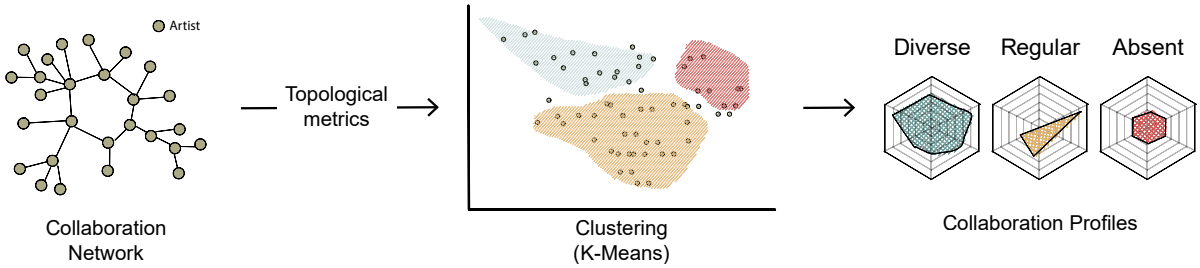


Figure 2. Overview of the collaboration-based feature creation process.

As already mentioned (Section 3.1), we define *hits* as songs that have appeared on the weekly Billboard Hot 100 chart at least once. An even more challenging task is to define negative (*non-hit*) samples as there is no data on less popular songs officially available. Besides, such songs—commonly referred to as *flops*—represent the majority of songs released, making any sample prone to bias. As an alternative, following Dewan and Ramaprasad; Singhi and Brown, we improved MusicOSet by collecting all the *hit-artists*' songs[3] that have never appeared on Hot 100 by using the artists' albums as crawling seeds over the Spotify API.[4] Finally, for a fair, more meaningful analysis, we consider only songs with one or two artists (which is 97% of the data, as shown in Figure 3; i.e., there is no loss of generality).[5] In practice, we call the leading artist on the song as *ego* and the featured artist as *alter*. The resulting dataset[6] contains 911,027 songs: 11,959 hits (1.3%) and 899,068 non-hits (98.7%).

To thoroughly understand the dataset, Figures 4 to 6 provide insightful descriptive information. Figure 4 presents the number of hits and non-hits according to the year of song release. The figure reveals that the number of non-hit is notably greater than that of hits, which is a reasonable expectation given the competitive nature of the music industry. Additionally, Figure 5 depicts the number of distinct albums and artists per song release year, showing a noticeable growth over the years. This trend suggests an increase in the number of new artists entering the industry, along with a growing diversity of music styles and genres. Finally, Figure 6 shows the number of artists per collaboration profile, where almost

[3]All the artists present in the original MusicOSet dataset.

[4]Spotify API: `https://developer.spotify.com/`

[5]Since we consider artist-based factors as one of the perspectives to predict the success of a song, such filtering facilitates data modeling and reduces the dimensionality of our multi-perspective models.

[6]For download, check the *Updates* link on `https://marianaossilva.github.io/DSW2019/`.

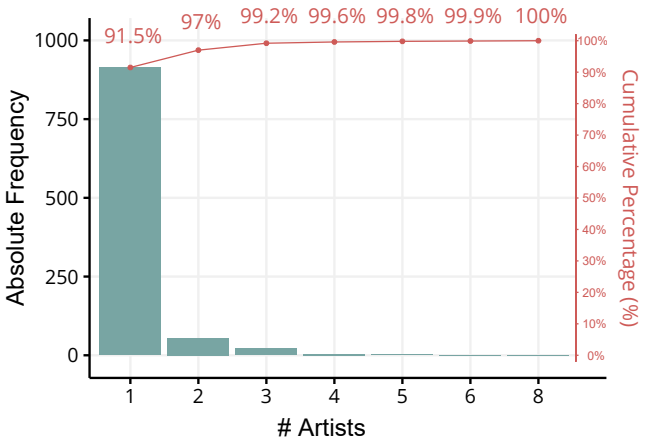


Figure 3. Pareto plot of the frequency and the cumulative percentage of the total number of artists on a song.

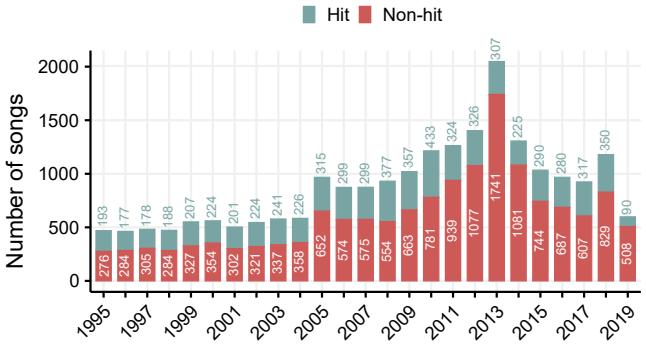


Figure 4. Number of hit and non-hit songs per song release year.

65% of artists present the *Absent* profile, i.e., they do not collaborate with other artists. Only a tiny percentage (1.2%) of artists fall under the *Diverse*, indicating that they collaborate with a wide range of other artists.

Correctly processing data through the learning models also requires handling different ranges and missing data for numeric and categorical features. Hence, we perform two-step numerical and categorical transformations. In the numerical transformation, only 0.28% of the songs have missing values, so we fill such values with the mean value for each numeric attribute. Although the mean imputation can distort the distribution for the missing variable, it works well with small numerical datasets and is the easiest and fastest way to impute missing values. Then, all attributes are normalized into a $[0, 1]$ range with the MinMax Scaler (Géron, 2019). In categorical transformation, for each categorical attribute, we fill missing values with a constant value, avoiding null problems. Finally, to adjust the data to the input format of most learning models, we binarize these features through the One-hot

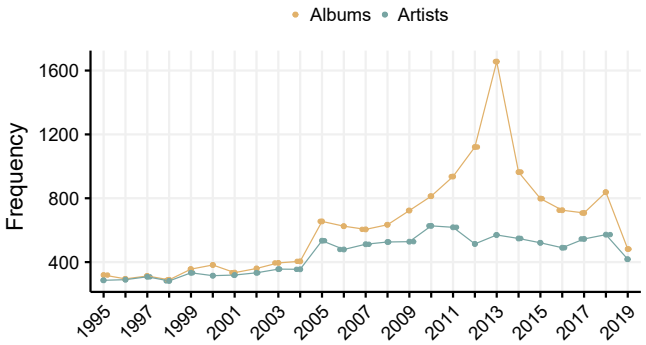


Figure 5. Number of distinct albums and artists per song release year.

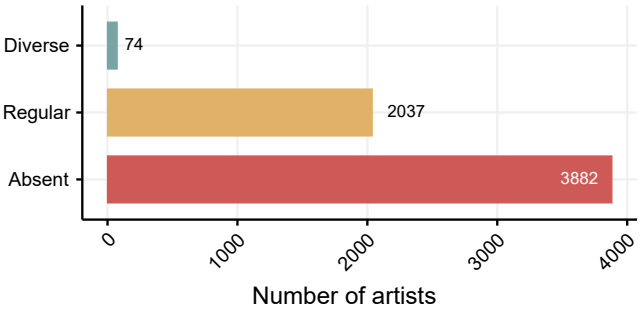


Figure 6. Number of artists per collaboration profile.

Encoding technique (Géron, 2019).

4.2 Classification Algorithms

To classify hit songs, inspired by previous works (Cosimato et al., 2019; Araujo et al., 2019), instead of choosing just one algorithm, we use different well-known classification methods to assess our solution performance. Specifically, we consider Random Forest and Gradient Boosting ensemble classifiers, Support Vector Machines (SVC and NuSVC), and the Multi-layer Perceptron (MLP) neural network model. We briefly describe each classifier, as follows.[7]

Ensemble methods. *Random Forest* is an ensemble approach that can be used to perform both classification and regression tasks. The algorithm combines several decision trees in randomly selected data samples to determine the final classification. Each decision tree is executed in parallel and, in the end, the algorithm selects the best solution through voting. *Gradient Boosting* is a generalization of boosting to arbitrary differentiable loss functions. It is an accurate and effective off-the-shelf procedure that can be used for both regression and classification problems in a variety of areas including Web search ranking and ecology.

Support Vector Machines. *Support Vector Classification* (SVC) is the classifier variant of SVM. It finds a hyperplane that best separates a multidimensional space into different classes based on the provided kernel function. Its main objective is to segregate the given dataset in the best possible way, by selecting a hyperplane with the maximum possible margin between support vectors in the given dataset. *NuSVC* is similar to SVC, but it accepts slightly different sets of parameters and has different mathematical formulations. It uses a parameter (nu) to control the number of support vectors.

Neural Networks. *Multilayer Perceptron* (MLP) learns a non-linear function approximator for either classification or regression. It differs from logistic regression as there can be one or more non-linear layers (*hidden layers*) between the input and the output layer.

4.3 Placement Algorithm

To address the hit song prediction as a *placement* task, we adapt the Learning to Place (L2P) algorithm (Wang et al., 2019) that learns to place a new instance into an ordinal list of known instances, ranked by a popularity measure. In the musical context, such measures can be sales profit, reputation on social media, or awards received. Here, we rely on

[7] We refer to related literature for complete definitions (Murphy, 2012).

Algorithm 1: Learning to Place Hit Songs (L2PHS)

Input: Training songs S, target variable vector t, test feature vector f_q and classifier C

1 $y = [\,]$ `# label vector`
2 $I = [\,]$ `# voting counter`
`# Training Phase`
3 **foreach** *pair of train songs* $(i, j) \in S \times S,\ i \neq j$ **do**
4 $X_{ij} = \texttt{ConcatenateFeatVector}(f_i, f_j)$
5 $y_{ij} = \texttt{CreatePairwisePreferences}(t_i, t_j)$
6 $C.\texttt{train}(X_{ij},\ y_{ij})$ `# train the model`
`# Testing Phase`
7 $intv = \texttt{sort}(\texttt{unique}(t))$ `# unique intervals`
8 **foreach** *test song* q **do**
9 **foreach** *train song* $i \in S$ **do**
10 $X_{iq} = \texttt{ConcatenateFeatVector}(f_q, f_i)$
11 $\hat{y}_{iq} = C.\texttt{predict}(X_{iq})$
12 $I = \texttt{Voting}(\hat{t}_{iq})$ `# voting process`
13 $h = \texttt{GetHighestInterval}(I)$ `# get the most voted interval`
14 $\hat{t}_q = \texttt{mean}(intv\,[h-1]\,,\ intv\,[h])$ `# get predicted place`
15 **return** $\hat{t}_q$

the Billboard Hot 100 chart, which is based on sales, radio airplay and streaming activity. Following the methodology in Lee and Lee (2018), we use an inverse-point system of the Billboard ranking: *rank_score* of song i is $rank\_score(i) = max\_rank - rank(i) + 1$, where *max_rank* is the lowest rank of the chart, and *rank(i)* is the song rank. Such an inverse-point system assigns higher scores to songs ranked higher in the chart. This can be useful for creating rankings or playlists based on popularity, as it gives more weight to the songs that are most popular among listeners. For example, if the chart has a maximum rank of 100 and a song is ranked at position 5, its *rank_score* would be $rank\_score(5) = 100 - 5 + 1 = 96$. This means the song would receive a high *rank_score* and be considered more popular than songs with lower *rank_scores*.

At first glance, one might assume that building a strong regression model is enough for predicting hit songs, as the target variable is a continuous measure. However, in general, traditional prediction and regression often struggle to accurately predict high-value instances of heavy-tailed distributions (Hsu and Sabato, 2016; Wang et al., 2019). A heavy-tailed distribution is one that is dominated by a large number of less popular items, with only a few very popular ones, such as in the case of creative industries like blockbuster movies (Collins et al., 2002), art auctions (Fraiberger et al., 2018), book sales (Wang et al., 2019), and, most relevantly, hit songs (Celma and Cano, 2008).

By definition, the L2P algorithm aims to estimate heavy-tailed outcomes and define performance measures for heavy-tailed target variables prediction. Although the *rank_score* (ranges in [1 to 100]) does not follow a heavy-tailed distribution, the songs' overall success distribution follows, as there are far more low-successful songs than high-successful songs. Therefore, we adapt the L2P algorithm to predict the position of a new song on a chart (given a sequence of previously ranked songs). Next, we briefly describe the L2P method, summarized by Algorithm 1.

Learning to Place Hit Songs. As a classical supervised learning method, L2PHS learns from a set of well-labeled input data and uses learned models to predict a quantitative outcome for a given test instance. It has two phases: *training*, which trains a classifier to predict pairwise preferences between each pair of training songs; and *testing*, which places a new song q in the given sequence of songs from the training set ranked by *rank_score*. A toy example is illustrated in Figure 7. As described in Algorithm 1, during training (lines 3-6), for each pair of songs $\{i, j\}$, L2PHS concatenates their feature vectors $\{f_i, f_j\}$ (line 4). Next, the problem "becomes" a binary classification based on the target variable with results: 1 or -1 (line 5). Then, L2PHS uses the training data as input to a classifier C to predict whether the *rank_score* for i is greater (or less) than j's. During testing (lines 7-14), each test song q is compared with each training song $i \in S$ using the model learned in the training phase to predict the pairwise relations (lines 8-11). Next, L2PHS treats each training song as a "voter". Training instances (voters) are sorted by their target variables, *rank_score*, in descending order, dividing the target variable axis into bins (line 12). If $\hat{y}_{iq} = 1$, bins on the right of t_i will obtain an *upvote* (+1), and bins on the left of t_i will obtain a *downvote* (-1). If $\hat{y}_{iq} = -1$, will *upvote* for bins on the left of t_i, and *downvote* for bins on the right of the t_i. After voting, L2PHS obtains a voting distribution over the bins. It then gets the most voted bin, h, and obtains the predicted place $\hat{t}_q$ as the midpoint of h (lines 13-14).

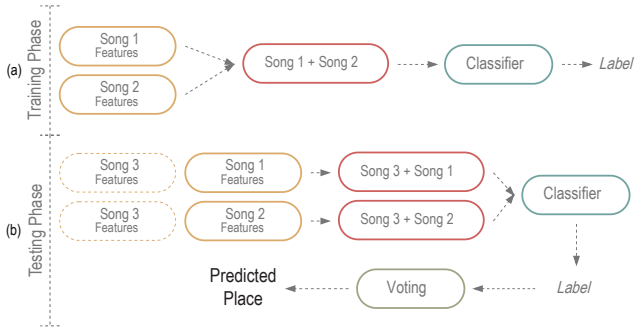


Figure 7. Learning to Place (L2P) adapted for music context. (a) *Training:* train a classifier on the pairwise relationship between each pair of train songs. (b) *Testing:* predict pairwise preferences between a new song, $Song\ 3$, and all train songs using the trained classifier; place $Song\ 3$ in the given sequence of trained songs ranked by *rank_score* through voting.

5 Experimental Evaluation

This section evaluates all five HSP models according to the task considered. First, we present the experimental setup and metrics to evaluate each evaluated task (Section 5.1) when comparing the performance of the HSP models (Section 5.2). Second, we investigate feature importance based on the models' results (Section 5.3).

5.1 Experimental Setup and Metrics

Binary Classification. We train and test each HSP variant model individually for fair evaluation with 75% of the data (chronologically split from 1995 on), leaving 25% for testing. As the train-test split follows a chronological order, we test the models against unseen data (e.g., whether a song released in 2020 will be a hit based on data up to 2019). That is, we investigate whether the models can predict future hit songs. Moreover, we set the parameters of all learning meth-

ods to their default values.[8] Here, we are more interested in investigating if collaboration features impact the models' performance rather than comparing them with a baseline while evaluating our models. To generate collaboration-based features, we consider the years of each partition individually: training, from 1995 to 75%, and test, all years of the dataset

To address the issue of class imbalance in our dataset, we employed a common strategy of randomly duplicating observations from the minority class (i.e., hits) with replacement in the training set. This is necessary since our dataset contains more non-hit songs (98.7%) than hit songs (1.3%), which could lead to biased model performance. By oversampling the minority class this way, we make the minority class equal to the majority, reinforce the hit song signal in the training set and improve the model's ability to predict them accurately. Note that the resampling is done only on the training set, or the performance measures could get skewed. The test set continues with a high imbalance level to mimic real-world data, where only a few songs can be considered hits.

When working with imbalanced classification, distinct evaluation metrics are often required. Unlike standard evaluation metrics that treat all classes equally important (e.g., accuracy), imbalanced classification typically rates classification errors by the minority class as more important than those by the majority class. Hence, we focus on the recall, precision, F1-Macro,[9] and AUC Score evaluation metrics to assess the best prediction model, as summarized next.

- *Recall (R):* measures how many of the positive cases the classifier correctly predicted over all the positive cases in the data. It is computed by the ratio $tp/(tp+fn)$ where tp is the number of true positives and fn the number of false negatives.
- *Precision (P):* measures how many of the positive predictions made are correct. It is computed by the ratio $tp/(tp+fp)$, where tp is the number of true positives and fp the number of false positives.
- *Area Under the ROC Curve (AUC-ROC):* measures the entire two-dimensional area underneath the entire ROC (Receiver Operating Characteristic) curve. The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.
- *F1-Macro (F1-M):* is the arithmetic mean of all the per-class F1 scores, without taking label imbalance into account. The F1-Score can be interpreted as a harmonic mean of the precision and recall, where the relative contribution of precision and recall is equal. It is computed by $2 * (precision * recall)/(precision + recall)$.

Hit Song Placement. We consider each final week's Billboard Hot 100 chart[10] of every 2018 month (the last completed year in our dataset). Therefore, we train 12 models for each HSP variant. Moreover, in our experiments, each model is trained and evaluated using the *Leave-One-Out* approach to split the data. The L2PHS algorithm is applied once for each data point, using all other songs as a training set and the selected instance as a test set (singleton). As discussed earlier (Section 4.3), L2PHS uses a binary classifier during the training phase. Here, we use a Random Forest classifier, as it has good performance (i.e., it does not overfit) and provides interpretability of features and results (Wang et al., 2019).

Finally, as evaluation metrics, we consider the following analyses: (*i*) *Quantile-Quantile plots* (Q-Q), the closer the values form a straight line, the higher chance to come from a similar distribution; and (*ii*) *Mean Absolute Error* (MAE) computes the average of the absolute difference between the actual and predicted values without considering their direction. As L2PHS predicted outcome is a continuous value, we use the MAE loss to compute the average absolute difference between the actual and predicted values without considering their direction. We chose this regression loss metric because it returns more easily interpretable errors and is not sensitive toward outliers.

Table 3. Binary classification performance evaluation, with classifiers sorted by F1-Macro.

Classifier	HSP Variant	R	P	AUC-ROC	F1-M
MLP	HSP-song-collab	0.810	0.677	0.899	0.810
	HSP-three-collab	0.757	0.693	0.818	0.804
	HSP-three	0.821	0.614	0.809	0.778
	HSP-song	0.782	0.574	0.777	0.747
	HSP-collab	0.508	0.438	0.699	0.619
SVC	HSP-song-collab	0.832	0.648	0.887	0.800
	HSP-three-collab	0.836	0.644	0.827	0.798
	HSP-three	0.870	0.576	0.809	0.762
	HSP-song	0.856	0.549	0.789	0.741
	HSP-collab	0.387	0.448	0.612	0.603
NuSVC	HSP-song-collab	0.823	0.637	0.887	0.792
	HSP-three-collab	0.827	0.636	0.821	0.792
	HSP-three	0.870	0.570	0.806	0.758
	HSP-song	0.858	0.556	0.794	0.746
	HSP-collab	0.277	0.511	0.590	0.591
GradientBoosting	HSP-song-collab	0.759	0.649	0.878	0.784
	HSP-three-collab	0.750	0.656	0.798	0.785
	HSP-three	0.778	0.611	0.792	0.768
	HSP-song	0.770	0.575	0.773	0.745
	HSP-collab	0.484	0.438	0.694	0.616
RandomForest	HSP-song-collab	0.590	0.745	0.881	0.772
	HSP-three-collab	0.580	0.769	0.761	0.775
	HSP-three	0.635	0.706	0.765	0.774
	HSP-song	0.580	0.643	0.727	0.734
	HSP-collab	0.445	0.445	0.676	0.614

5.2 Performance Comparison

Binary Classification. Table 3 presents the performance of the five classifiers ordered by F1-Macro. For all classifiers, the HSP-three-collab variant presented the best F1-Macro values, reinforcing the hypothesis that considering collaboration-based features increase the predictive power of such classifiers. The best-performing classifier for all variants is the MLP (Multi-layer Perceptron), with a F1-Macro ranging from 75% to 80%. Its precision and accuracy also present high values, making it a good classification method, regardless of the model considered. In binary classification, precision and recall are both essential metrics, but in the context of hit song prediction, precision is considered more im-

[8] All considered classifiers were trained using the python *scikit-learn* library. The default parameters of each algorithm can be obtained from the documentation: `https://scikit-learn.org/stable/supervised_learning.html#supervised-learning`

[9] For imbalanced datasets, the macro-average gives every class the same importance and, therefore, better reflects how well the model performs.

[10] The Billboard Hot 100 chart is released weekly on Tuesdays. It lists the 100 currently popular songs across all genres, ranked by sales (physical and digital), radio play, and online streaming in the United States.

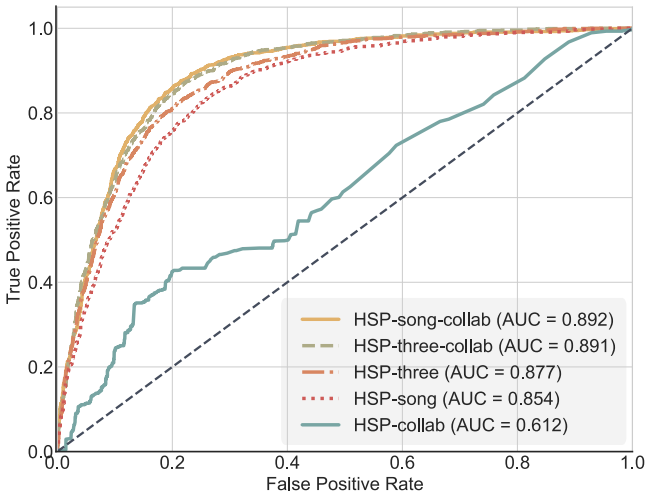


Figure 8. ROC curve performance measurement and area under the curve (AUC) score of the MLP classifier for each HSP model. Each model is indicated with a different color and shape.

Table 4. *Hit Song placement* performance evaluation for all months.

Month	*MAE (Mean Absolute Error)*				
	HSP-s	**HSP-t**	**HSP-c**	**HSP-s-c**	**HSP-t-c**
January	0.86 ± 0.18	0.87 ± 0.20	0.40 ± 0.17	0.41 ± 0.16	0.42 ± 0.16
February	0.89 ± 0.21	0.82 ± 0.22	0.44 ± 0.15	0.52 ± 0.17	0.50 ± 0.17
March	0.67 ± 0.14	0.68 ± 0.16	0.39 ± 0.13	0.44 ± 0.13	0.46 ± 0.13
April	0.74 ± 0.17	0.64 ± 0.16	0.53 ± 0.17	0.47 ± 0.15	0.46 ± 0.14
May	0.74 ± 0.15	0.74 ± 0.16	0.50 ± 0.12	0.57 ± 0.14	0.55 ± 0.13
June	0.68 ± 0.17	0.62 ± 0.15	0.52 ± 0.15	0.54 ± 0.16	0.52 ± 0.15
July	0.63 ± 0.17	0.62 ± 0.17	0.53 ± 0.18	0.48 ± 0.17	0.50 ± 0.16
August	0.72 ± 0.15	0.75 ± 0.15	0.51 ± 0.13	0.47 ± 0.13	0.47 ± 0.13
September	0.66 ± 0.18	0.68 ± 0.18	0.50 ± 0.17	0.51 ± 0.15	0.54 ± 0.16
October	0.72 ± 0.16	0.71 ± 0.16	0.45 ± 0.13	0.47 ± 0.14	0.48 ± 0.13
November	0.68 ± 0.20	0.68 ± 0.20	0.35 ± 0.13	0.44 ± 0.15	0.45 ± 0.16
December	0.68 ± 0.13	0.70 ± 0.15	0.31 ± 0.11	0.38 ± 0.12	0.40 ± 0.12

portant than recall. This is because the goal is to identify hit songs accurately, and false negative predictions are more detrimental than false positives. Thus, prioritizing high precision, even if it leads to lower recall, is more beneficial. The MLP classifier's high precision and AUC Score indicate its suitability as a classification method for hit song prediction, regardless of the model considered. Additionally, we evaluate the ROC curves of the MLP classifier for each variant to provide a more comprehensive performance analysis.

Figure 8 depicts the ROC curves and AUC scores for each HSP model, trained using the MLP classifier. The ROC AUC measures the model's ability to distinguish between hit and non-hit songs. The higher the AUC score, the better the model can differentiate between these two classes. As shown in the figure, the HSP-collab variant had the poorest performance, which is expected since classifying hit songs based solely on the collaborative characteristics of the artists is not reasonable. Moreover, the variants that consider factors other than song-based ones outperform the **HSP-song** model, which considers only the *song* perspective. Although the **HSP-three**, **HSP-three-collab**, and **HSP-song-collab** models had similar results, the latter, which takes song and collaboration features into account, was the best model among the five investigated.

Hit Song Placement. We assess the expected *rank_score* against observed outcomes for each model variant along all months (Figure 9). In Q-Q plot analysis, best performing models are close to the line $x = y$ (i.e., at 45°). Moreover, the Q-Q points form a line if the distributions are linearly related but not necessarily at $x = y$. Overall, the **HSP-collab** variant is the closest to the ground truth at 45°, followed by the other two models that include collaboration-based features (**HSP-three-collab** and **HSP-song-collab**). On the other hand, the two models that do not include collaboration features, **HSP-song** and **HSP-three**, systematically underpredict the actual *rank_score* and show a deviation at the upper and lower ends. However, the **HSP-collab** model produces the smallest deviation, indicating that incorporating collaboration characteristics can improve the model's performance.

Despite the relevant results, comparing the deviations between both distributions is insufficient to evaluate the models by themselves, as the error between predicted and actual *rank_score* is not directly gauged by such metrics. Hence, we calculate the MAE regression loss metric to quantify the models' performance using prediction errors. Table 4 summarizes the computed mean absolute errors. As expected from the Q-Q plot analysis, the **HSP-collab** variant achieves the lowest MAE for most months, indicating the best predictive performance. In contrast, the **HSP-three** and **HSP-song** models without collaboration-based features show higher MAE values, indicating poorer predictive performance. The models incorporating collaboration-based features (**HSP-three-collab** and **HSP-song-collab**) also outperform the non-collaborative models but show slightly higher MAE values than the **HSP-collab** model for most months.

General Points. Overall, the variants with collaboration-based features demonstrated superior performance across all studied scenarios and tasks. The performance comparison results also indicate that considering predictive factors from different perspectives, in addition to the song perspective, can improve the effectiveness of hit song prediction models. Although, the best performance is not related to including additional features (i.e., increasing the number of perspectives considered in the model) since the **HSP-three** model, which considers features from all three perspectives, achieved similar performance and, in some cases (mainly in the *hit song placement* task), inferior to the variant that only considers song-based features. Such results indicate that considering features based on artists' collaborations can improve the models' performance in distinguishing hit from non-hit songs. Specifically, based on the F1-Macro and average MAE of the *HSP-song* model as a baseline, considering artists' collaboration features increased 8.4% the F1-Macro and decreased 34.3% the average MAE, which answers our first research question (**RQ1**). In the next section, we delve into the specific factors that have the most significant influence on hit song prediction.

5.3 Feature Importance

Binary Classification. Machine learning algorithms can produce good predictions, but their *black-box* nature does not help in understanding highly trained models. However, understanding how features influence prediction is still relevant. Hence, we use the SHAP (SHapley Additive exPlanations) (Lundberg and Lee, 2017) values, a unified framework of feature importance to interpret predictions by

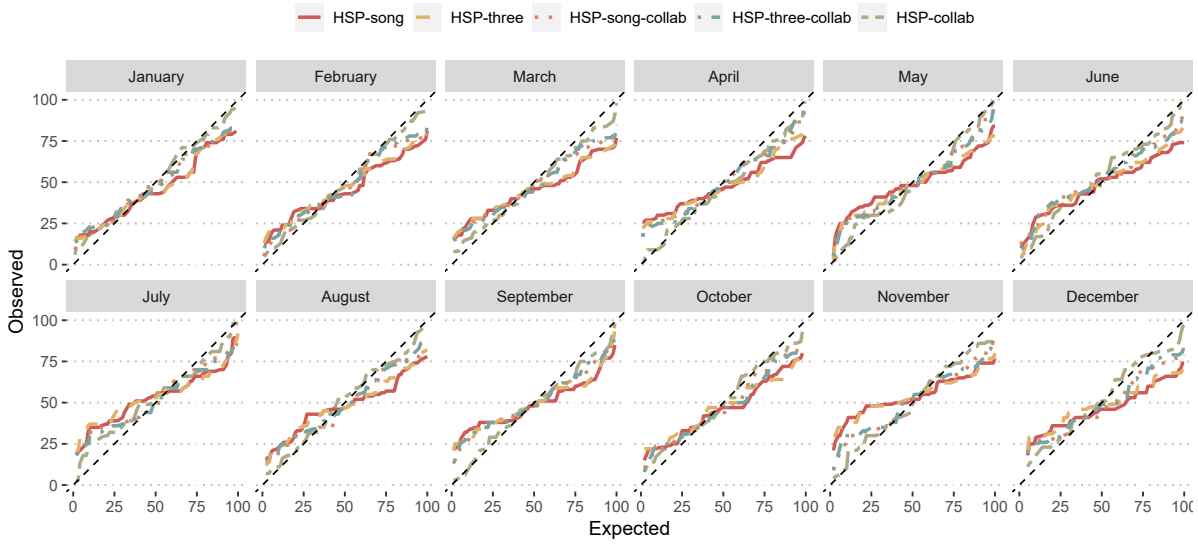


Figure 9. Quantile-Quantile (Q-Q) plots for all 2018 months, comparing the observed outcomes with the expected distribution for each HSP model. Each model is indicated with a different color and line type. The diagonal dashed line indicates identity.

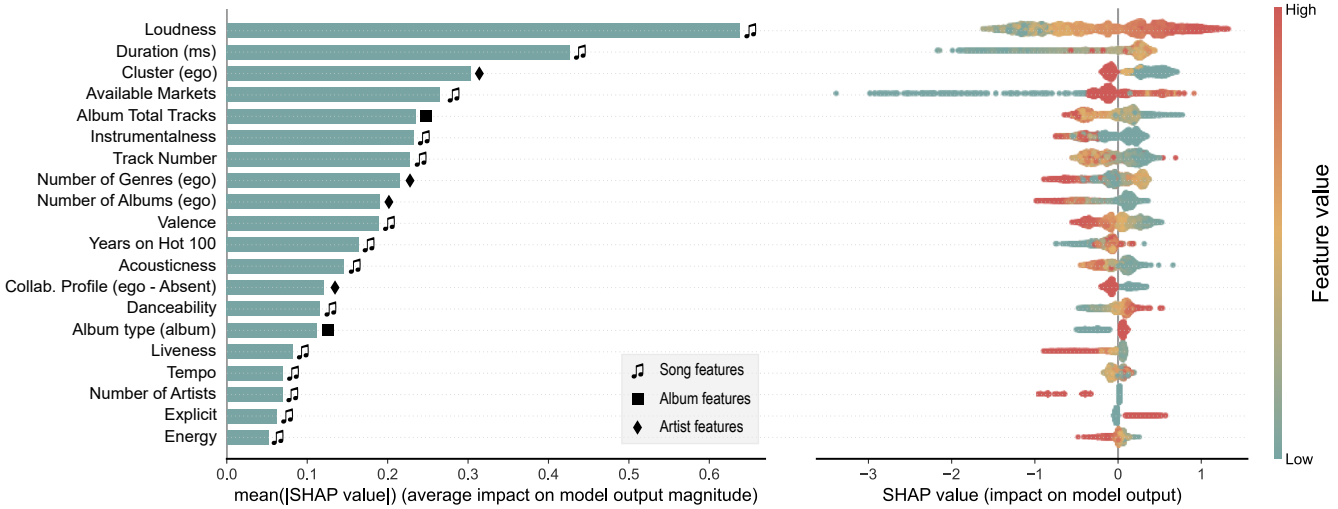


Figure 10. Top 20 most significant variables. Left: relative importance as the mean absolute Shapley values. To better represent the features type, ♫ marks the song features, ♦ the artist features, and ■ the album features. Right: SHAP summary plot of the hit prediction model (the horizontal location shows whether the impact of a feature is associated with a higher or lower prediction, while the colors indicate high/red or low/green feature value).

computing the contribution of each feature to the results for *GradientBoosting* learning algorithm.[11]

The global importance of features included in **HSP-three-collab** is illustrated in Figure 10 by summary plots, where all features are vertically sorted by their average impact in the predictions. The feature importance plot (left) is useful, but there is no information beyond the relative importance. The summary plot (right) can further show the positive and negative relationships of the predictors with the target variable, combining feature importance with feature effects. Each point indicates a Shapley value for a feature and an instance. The feature determines the position on the y-axis and on the x-axis by the Shapley value, i.e., the impact that feature has on the model's prediction for that song.

Also, Figure 10 (right) reveals the direction of feature effects, such as explicit songs (red) having a high and positive impact on the quality rating (the *high* is in red, and the *positive* impact is shown on the x-axis). Such behavior is consistent with the expected and illustrated by Figure 11. More than 50% of Billboard Hot 100 #1 songs in 2018 feature explicit lyrics. The taste for expletive-filled lyrics has grown since 2012, except for 2017. From 2014 on, at least 20% of number one songs have the label of parental advice.

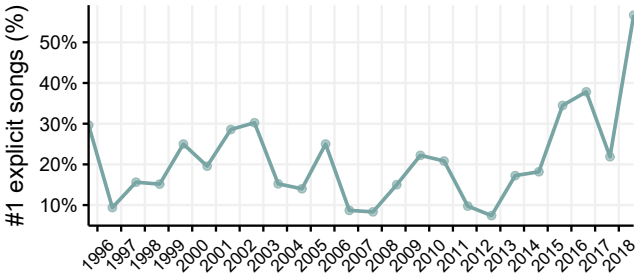


Figure 11. Presence of average explicit lyrics in Billboard Hot 100 #1 hit songs in 1995–2018. From 2014 on, at least 20% of top-chart songs have parental advice labels.

Additionally to the direction, the summary plot (Figure 10 right) provides the distribution of effect sizes, such as the

[11] We used the GradientBoosting algorithm instead of the Multilayer Perceptron (MLP) because there is no support for the MLP algorithm in the SHAP framework.

long tails of some features. The general trend of long tails reaching to the left but not to the right means that extreme values of such measurements can significantly raise non-hit prediction. It also means that features with low global importance (e.g., *ego_num_albums* and *liveness*) can still be necessary for specific instances.

Overall, the summary plot emphasizes the relationship between a feature value and its impact on the prediction. As expected, most musical features are present in the top 20 of global importance, with *loudness* and *duration_ms* (i.e., track duration in milliseconds) having the maximum impact on the quality rating. Also, both features presented similar effects, with high values associated with a positive impact on hit song prediction. In acoustics, *loudness* represents the subjective perception of sound pressure and is directly proportional to the square of the vibration amplitude. This is compatible with previous intuitions and scientific knowledge. According to (Ni et al., 2011), there is an evident trend for music to become relatively longer and louder. Hence, the increasing importance of such metrics has become more useful for telling apart a hit from a non-hit.

For the *album* perspective, there are two features among the most significant predictors: low values of *album_total_tracks* and *album_type_album* feature equal to 1 (i.e., when the album has seven tracks or more), the hit song predictions increases. However, the global importance of both features is quite different, with the number of tracks on a song's album being much more significant than that song being released within an album. In other words, hit songs tend to appear on albums composed of few songs. In the music industry, albums released with one to three tracks are called *singles*. A *single* is frequently a song considered commercially viable enough by the artist and the recording company to be released separately from an album. Hence, the result is consistent with such a reality.

Regarding the *author*'s perspective, *ego_cluster* is the third most important predictor, changing the predicted absolute hit probability on average by 30% percentage points. Note that the artist's collaborative feature (*Collab. Profile*) significantly affects the model's accuracy for predicting successful songs. This artist-based feature indicates which cluster an artist is a part of, i.e., which is his/her collaboration profile: Diverse, Regular and Absent. Note that the Absent profile is among the top 20 most influential features. As Figure 10 (right) suggests, an artist with such a profile (i.e., *ego_profile_Absent* = 1) negatively drives the predictive model to the *non-hit* class; likewise, when equal to 0, the corresponding Shapley values are positive. This means the collaborative information of the artist significantly affects the model's accuracy for predicting successful songs, especially when artists do not have a non-collaborative profile, i.e., belonging to Diverse or Regular profiles.

Finally, the results indicate that collaboration features, especially the artist's collaboration profile, are critical for accurately predicting song success. This finding is supported by the consistent feature importance results obtained when comparing different subsets of input features (Table 3). Specifically, when the subset of input features without collaboration features as compared to the complete set of input features, the latter consistently outperformed the former in predictive power. Therefore, including collaboration features in the model is essential for accurately predicting song success.

Hit Song Placement. Following Wang et al. (2019), given our three perspectives (song, album and artist), we now assess their relative importance by training three individual models (i.e., each model uses only the features of one perspective). We use all models separately to predict the *rank_score* of each song, applying the L2PHS algorithm. We compare them to the actual *rank_score* of the songs and normalize the absolute errors E_{music}, E_{artist} and E_{album} to sum them up to one. Then, we use a ternary plot to inspect the source of errors for songs.

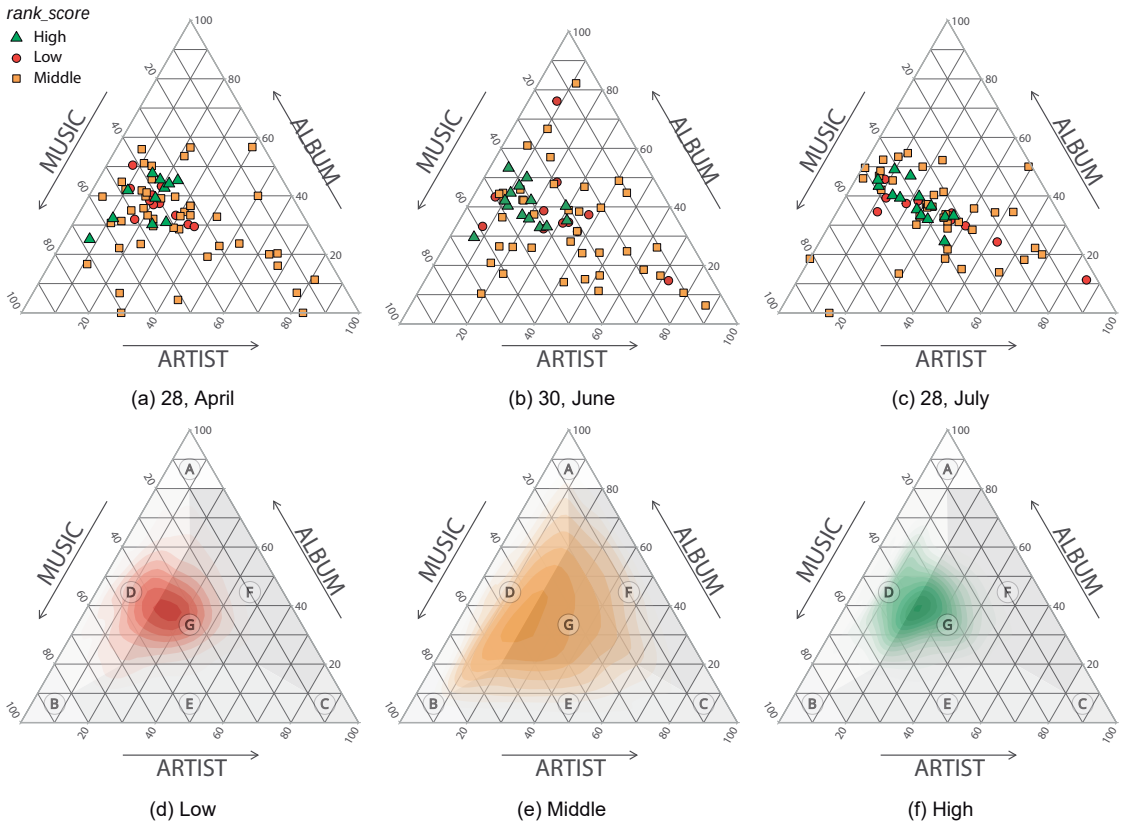
Figures 12 (a-c) show ternary plots in three selected months (April, June, and July).[12] We color each data point and set different shapes based on actual *rank_scores*. Figures 12 (d-f) show the normalized absolute error accumulated in a year per *rank_score* level. In summary, the maximum E_{music} value is in the left bottom corner of the triangle, the maximum E_{artist} is at the right bottom, and the maximum E_{album} is at the top. Overall, the central left side (Region **D**) has the highest density of absolute errors for songs with high or low *rank_score*. For songs with average *rank_score*, they are more distributed in the central region of the triangle (Region **G**), with some outliers concentrated in the lower right corner (Region **C**).

The data concentration in Region **D** indicates that relying only on *song* and *album* perspectives returns the most significant prediction error; i.e., considering only *song* and *album* information is not enough for a good prediction. Songs with a medium to high *rank_score* are also concentrated in Region **G**, which indicates good hit song placement requires excelling in all three perspectives: *song*, *artist* and *album*. Finally, there is a small concentration of outlier songs with average popularity in the right corner (Regions **E**, **F** and **C**), which means that having only *artist* information can negatively affect model performance for isolated cases.

General Points. From the feature importance analysis, one may think that the resources with the highest scores are enough to predict hit songs with high efficiency. Such filtering would not only reduce the dimensionality of the model, making it simpler but also speed up its operation, possibly improving the final performance. Therefore, to assess whether the three most important features, according to SHAP values, are sufficient for a good prediction, we trained an additional model (**HSP-top3**) that considers only the top three features: *loudness*, *duration_ms* and *ego_cluster*. We used the MLP classifier, which resulted in an F1-Macro of 0.56 (Precision: 0.37; Recall: 0.71; AUC Score: 0.65), much lower than the results of the previous variants, which means that relying on these three factors alone to predict a hit song is not enough.

Finally, we also evaluated another model (**HSP-collab**) to investigate whether considering only collaboration features is enough for a good prediction. Here, we also use the MLP classifier, which resulted in an F1-Macro of 0.63 (Precision: 0.46; Recall: 0.49; AUC Score: 0.70). Although the model resulted in a good performance, such results reinforce the idea

[12] Due to space limitations, we selected the months in which the HSP-collab model presented the highest MAE values, i.e., the worst performances.



General explanation. The values of the three perspectives *song*, *artist* and *album* sum 100%. The concentration of each perspective is 100% in each corner of the triangle and 0% at its opposite line. Besides its three corners, the ternary plot can be divided into seven regions: (**A**) contains at least 80% of E_{album}; (**B**) contains at least 80% of E_{song}; (**C**) contains at least 80% of E_{artist}; (**D**) contains no more than 20% of E_{artist}; (**E**) contains no more than 20% of E_{album}; (**F**) contains no more than 20% of E_{song}; and (**G**) contains at least 20% of each perspective.

Figure 12. Ternary diagram plots for feature importance. (a, b, c) show the normalized absolute error for feature group importance ($E_{song}, E_{artist}, E_{album}$) in three selected months. For each data point, the three values are the normalized absolute error generated by L2P with only the corresponding feature group. (d, e, f) show the normalized absolute error accumulated in 2018 per *rank_score* level. The *rank_score* is divided into three categories: *low*, *middle*, and *high*, corresponding to popularity metric within $[0, 20]$, $(20, 80]$, and $(80, 100]$, respectively. Capital letters (A to G) represent seven regions of each ternary plot, as described by **General explanation** above.

that a multi-perspective approach is essential to tackle the Hit Song Prediction problem, especially considering features based on artist collaboration. Overall, both feature importance analyses answer our second research question (**RQ2**), showing that the *artist* perspective strongly contributes to the hit song prediction problem and considering factors from multiple perspectives of the musical context positively impacts the prediction model outcome.

6 Conclusion

In this paper, we addressed the problem of Hit Song Prediction, which is common in *Hit Song Science*. Here, we define this problem from two different tasks: *binary classification* and *hit song placement*. To tackle both tasks, we evaluated hit song prediction (HSP) models, which consider different factors from three perspectives: *song*, *artist* and *album*. We designed five HSP model variants (**HSP-song**, **HSP-three**, **HSP-collab**, **HSP-song-collab** and **HSP-three-collab**) to assess not only the effect of relying only on song-based features, but also the impact of including different perspectives (i.e., album and artists), mainly collaboration-based features.

Our proposed methodology sheds light on two RQs: (**RQ1**) by comparing the performance of the hit song prediction models (with different perspective factors), we found that relying exclusively on internal musical features is not enough to obtain effective hit song predictions, instead, considering three perspectives and collaboration-based features increases the prediction models' performance; (**RQ2**) the feature importance analyses allowed to identify the most significant features that drive hits prediction, and the *artist* modality contains the most influential predictors, mainly social interaction data. Such results show the relevance of handling Hit Song Prediction as a multi-perspective problem and the importance of relying on information from the artists' collaboration profiles. Furthermore, our results reveal that it is possible to predict whether a given song will be a hit.

Overall, our work differs from the current state of the art in two crucial ways. First, although considering artists' collaboration aspects in hit song prediction is beneficial from the analytical perspective, this is the first time the collaboration between artists and their profiles are modeled as features for a machine learning approach. Second, the multi-perspective approach brings the necessary complexity to analyzing music in many of its facets. Therefore, combining the multi-perspective representation with a collaboration-aware model

means a big step toward advancing both *Hit Song Science* and *Music Information Retrieval* fields, consequently providing a potential impact on the music industry.

Limitations and Future Work. One limitation of our work is its dataset comprising music charts from the U.S. only. Hence, a natural extension considers data beyond the U.S., such as European, Latin American, and Asian charts. Still, collecting and preparing such data for all tasks performed here presents severe challenges, including the lack of open online information. As future work, we also plan to include other interactions in social media in our multi-perspective approach and other characteristics in this context, such as artist reputation.

A second limitation is the data-oriented algorithmic nature of the problem definition and proposed solution. In other words, we take an algorithmic stance on an art-oriented context with strong emotional appeal, music. To do so, the only way is to limit the context and the variables that have power over it: a hit song is one that appears on Billboard Top 100, whose placement is defined by the features previously discussed. Anyone may question that other variables certainly play important roles in making such ranking, including marketing strength, timing, and other unforeseen interference (e.g., Kate Bush's 1985 hit "Running Up That Hill" sprints up the charts, thanks to Netflix show "Stranger Things").[13] Although it is a fair statement, our goal is to extract knowledge from known available data in order to predict a potential hit song, and our experimental evaluation shows the solution works given all definitions and closed scenario presented here.

Finally, using the default parameter values for the algorithms considered in the binary classification task may not produce the best possible results and lead to a bias regarding the most effective classifier. However, we believe that our analyses are still valid since our goal is not to present the best model for Hit Song Prediction, but rather to investigate whether collaboration-based features increase the prediction models' performance. Nevertheless, we still plan to perform a Grid Search in all algorithms to find the best parameter values and enhance the classification results.

Acknowledgements

This work was supported by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Brazil. The authors also like to thank Prof. Anisio Lacerda for all relevant insights to model and build this dataset.

References

Almada, C. et al. (2019). J-analyzer: A software for computer-assisted analysis of antônio carlos jobims songs. In *SBCM*, pages 12–16, Brazil. SBC.

Araujo, C. V., de Cristo, M. A. P., and Giusti, R. (2019). Predicting music popularity using music charts. In *ICMLA*, pages 859–864, Boca Raton, Florida, USA. IEEE.

Araujo, C. V. et al. (2017). Predicting music success based on users' comments on online social networks. In *WebMedia*, pages 149–156, Brazil. SBC.

Bischoff, K. et al. (2009). Social knowledge-driven music hit prediction. In *Advanced Data Mining and Applications*, pages 43–54, Berlin, Heidelberg. Springer.

Calefato, F., Iaffaldano, G., and Lanubile, F. (2018). Collaboration success factors in an online music community. In *Proceedings of the ACM Conference on Supporting Groupwork*, pages 61–70, Sanibel Island, USA. ACM.

Celma, Ò. and Cano, P. (2008). From hits to niches? or how popular artists can bias music recommendation and discovery. In *Netflix-KDD Work.*, pages 1–8.

Collins, A., Hand, C., and Snell, M. C. (2002). What makes a blockbuster? economic analysis of film success in the united kingdom. *Managerial and Decision Economics*, 23(6):343–354.

Cosimato, A. et al. (2019). The conundrum of success in music: Playing it or talking about it? *IEEE Access*, 7:123289–123298.

Costa, W. d. L., Filgueira, D., Ananias, L., Barioni, R., Figueiredo, L. S., and Teichrieb, V. (2020). Songverse: a digital musical instrument based on virtual reality. *Journal on Interactive Systems*, 11(1):57–65.

da Silva, A. C. M., Silva, D. F., and Marcacini, R. M. (2020). 4mula: A multitask, multimodal, and multilingual dataset of music lyrics and audio features. In *WebMedia*, pages 145–148, Brazil. ACM.

de Almeida, M. A. et al. (2017). The fast and winding roads that lead to the doors: Generating heterogeneous music playlists. In *WebMedia*, pages 269–276, Brazil. ACM.

Dewan, S. and Ramaprasad, J. (2014). Social media, traditional media, and music sales. *Mis Quarterly*, 38(1).

Dhanaraj, R. and Logan, B. (2005). Automatic prediction of hit songs. In *ISMIR*, pages 488–491, London, UK. Int'l Society for Music Information Retrieval.

Fraiberger, S. P. et al. (2018). Quantifying reputation and success in art. *Science*, 362(6416):825–829.

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, USA.

Hsu, D. J. and Sabato, S. (2016). Loss minimization and parameter estimation with heavy tails. *J. Mach. Learn. Res.*, 17:18:1–18:40.

Interiano, M. et al. (2018). Musical trends and predictability of success in contemporary songs in and out of the top charts. *Royal Society open science*, 5(5):171274.

Kim, S. T. and Oh, J. H. (2021). Music intelligence: Granular data and prediction of top ten hit songs. *Decis. Support Syst.*, 145:113535.

Kim, Y., Suh, B., and Lee, K. (2014). # nowplaying the future billboard: mining music listening behaviors of twitter users for hit song prediction. In *SoMeRA*, pages 51–56, Gold Coast, Australia. ACM.

Lee, J. and Lee, J. (2018). Music popularity: Metrics, charac-

[13] `https://www.npr.org/2022/06/07/1103445453/running-up-that-hill-kate-bush-stranger-things`, June 7, 2022

teristics, and audio-based prediction. *IEEE Transactions on Multimedia*, 20(11):3173–3182.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *NIPS*, page 4768–4777, Long Beach, California, USA. Curran Associates Inc.

Martín-Gutiérrez, D. et al. (2020). A multimodal end-to-end deep learning architecture for music popularity prediction. *IEEE Access*, 8:39361–39374.

Martins, G., Gomes, G., Conceição, J. L., Marques, L., da Silva, D., Castro, T., Gadelha, B., and de Freitas, R. (2021). Bumbometer digital crowd game: collaboration through competition in entertainment events. *Journal on Interactive Systems*, 12(1):294–307.

Murphy, K. P. (2012). *Machine learning - a probabilistic perspective*. MIT Press, Cambridge, USA.

Ni, Y. et al. (2011). Hit song science once again a science? In *Intl. Workshop on Mach. Learn. and Music*, Sierra Nevada, Spain. NIPS.

Nunes, J. C. and Ordanini, A. (2014). I like the way it sounds: The influence of instrumentation on a pop song's place in the charts. *Musicae Scientiae*, 18(4):392–409.

Pachet, F. (2011). Hit song science. In Li, T., Ogihara, M., and Tzanetakis, G., editors, *Music Data Mining*, chapter 10, pages 305–326. CRC Press, USA.

Pachet, F. and Roy, P. (2008). Hit song science is not yet a science. In *ISMIR*, pages 355–360, Philadelphia, USA. Int'l Society for Music Information Retrieval.

Ren, J., Shen, J., and Kauffman, R. J. (2016). What makes a music track popular in online social networks? In *WWW*, pages 95–96, Montreal, Canada. ACM.

Ren, L. et al. (2010). Dynamic Nonparametric Bayesian Models for Analysis of Music. *Journal of the American Statistical Association*, 105:458–472.

Roads, C. (1996). *The Computer Music Tutorial*. MIT Press, Cambridge, England.

Salganik, M. J. et al. (2006). Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, 311(5762):854–856.

Schedel, M. and Young, J. P. (2005). Editorial. *Organised Sound*, 10(3):181–183.

Serra, A. C. et al. (2021). Quality enhancement of highly degraded music using deep learning-based prediction models for lost frequencies. In *WebMedia*, pages 205–211, Brazil. ACM.

Serrà, J. et al. (2012). Measuring the evolution of contemporary western popular music. *Scientific Reports*, 2(521).

Silva, M. O. and Moro, M. M. (2019). Causality Analysis Between Collaboration Profiles and Musical Success. In *WebMedia*, pages 369–376, Rio de Janeiro. ACM.

Silva, M. O., Mota, L., and Moro, M. M. (2019a). MusicOSet: An Enhanced Open Dataset for Music Data Mining. `https://doi.org/10.5281/zenodo.4904639`.

Silva, M. O., Oliveira, G. P., Seufitelli, D. B., Lacerda, A., and Moro, M. M. (2022). Collaboration as a driving factor for hit song classification. In Silva, T. H., Dorini, L. B., Almeida, J. M., and Marques-Neto, H. T., editors, *WebMedia '22: Brazilian Symposium on Multimedia and Web, Curitiba, Brazil, November 7 - 11, 2022*, pages 66–74. ACM.

Silva, M. O., Rocha, L. M., and Moro, M. M. (2019b). Collaboration Profiles and Their Impact on Musical Success. In *SAC*, pages 2070–2077, Limassol, Cyprus. ACM.

Silva, M. O., Rocha, L. M., and Moro, M. M. (2019c). Musicoset: An enhanced open dataset for music data mining. In *XXXII Simpósio Brasileiro de Banco de Dados: Dataset Showcase Workshop, SBBD 2019 Companion*, pages 8–17, Fortaleza, CE, Brazil. SBC.

Singhi, A. and Brown, D. G. (2015). Can song lyrics predict hits. In *Proceedings of the 11th International Symposium on Computer Music Multidisciplinary Research*, pages 457–471.

Vötter, M. et al. (2021). Novel datasets for evaluating song popularity prediction tasks. In *IEEE International Symposium on Multimedia (ISM)*, pages 166–173, Los Alamitos, USA. IEEE.

Wang, X. et al. (2019). Success in books: predicting book sales before publication. *EPJ Data Science*, 8(1):31.

Yang, L., Chou, S., Liu, J., Yang, Y., and Chen, Y. (2017). Revisiting the problem of audio-based hit song prediction using convolutional neural networks. In *ICASSP*, pages 621–625, New Orleans, USA. IEEE.

Zangerle, E., Vötter, M., Huber, R., and Yang, Y. (2019). Hit song prediction: Leveraging low- and high-level audio features. In *ISMIR*, pages 319–326, Delft, Netherlands. Int'l Society for Music Information Retrieval.