Revista de Informática Teórica e Aplicada - RITA - ISSN 2175-2745
Vol. 27, Num. 04 (2020) 108-117

RESEARCH ARTICLE

A Model for Predicting Music Popularity on Streaming Platforms

Um Modelo para Previsão da Popularidade de Músicas em Plataformas de *Streaming*

Carlos V. S. Araujo[1]*, Marco A. P. Cristo[1], Rafael Giusti[1]

Abstract: The global music market moves billions of dollars every year, most of which comes from streaming platforms. In this paper, we present a model for predicting whether or not a song will appear in Spotify's Top 50, a ranking of the 50 most popular songs in Spotify, which is one of today's biggest streaming services. To make this prediction, we trained different classifiers with information from audio features from songs that appeared in this ranking between November 2018 and January 2019. When tested with data from June and July 2019, an SVM classifier with RBF kernel obtained accuracy, precision, and AUC above 80%.
Keywords: music — hit song science — machine learning — Spotify

Resumo: O mercado musical global movimenta bilhões de dólares todos os anos. A maioria desses bilhões vem de plataformas de *streaming*. Neste artigo, apresentamos um modelo que prevê se uma música irá ou não aparecer no Top 50 do Spotify, um ranking das 50 músicas mais populares nessa plataforma, que é um dos maiores serviços de *streaming* atualmente. Para fazermos essa previsão, nós treinamos diferentes classificadores com informações de características acústicas das músicas que apareceram nesse ranking entre novembro de 2018 a janeiro de 2019. Quando fizemos testes em dados de junho e julho de 2019, um classificador SVM com kernel RBF obteve acurácia, precisão e AUC superiores a 80%.
Palavras-Chave: música — *hit song science* — aprendizagem de máquina — Spotify

[1] *Institute of Computing, Federal University of Amazonas, Manaus, Brazil*
*__Corresponding author__: vicente@icomp.ufam.edu.br
DOI: http://dx.doi.org/10.22456/2175-2745.107021 • **Received:** 30/08/2020 • **Accepted:** 23/10/2020

1. Introduction

The way people listen to music is changing. In 2018, for the first time, streaming became the main form of music consumption, accounting for 47% of the music market, according to the International Federation of the Phonographic Industry (IFPI) annual report[1]. In 2019 this percentage is even higher accounting for 56.1% of global music revenues[2]. Therefore, streaming has become critical for artists and record labels to achieve good business results.

One way to help artists and record labels maximize commercial return is to use a model to predict whether their music will be popular on streaming platforms. To get a sense of the commercial impact that such a model could present, it is sufficient to say that the music market, considering only the revenue from music consumption and licensing, moved US$ 20.2 billion around the globe in 2019, according to the IFPI. In addition, this market is extremely competitive. As an example of this competitiveness, Wikipedia catalogs 1,396 music labels only in the United States[3]. Note that, according to the Data Usa website, there are 138,000 artists on American lands[4]. A prediction model could give artists and labels an edge over competitors, because they could focus more on songs that tend to earn a good yield.

Our work can be classified as a Hit Song Science (HSS) research. HSS studies ways to predict the success of songs before they are even available on the market. Therefore, it is an important area for artists and music labels to plan actions that can achieve greater financial return [1]. HSS is a sub-area of Music Information Retrieval (MIR), a research field whose focus is to gather information from songs [2].

In this paper we present an HSS model to predict if a song will be popular on the Spotify streaming platform. Spotify was chosen as our study case because it is the world's second largest music streaming service in number of users. The first one is Soundcloud, which lacks songs from renowned artists

[1] ⟨https://www.ifpi.org/downloads/GMR2019.pdf⟩
[2] ⟨https://www.ifpi.org/wp-content/uploads/2020/07/Global_Music_Report-the_Industry_in_2019-en.pdf⟩
[3] ⟨http://bit.ly/2sJ3dfE⟩. Access at 2020-08-13.
[4] ⟨https://datausa.io/profile/soc/272040/⟩. Access at 2020-08-13.

and record labels[5,6]. We consider a music to be popular if it has been featured in the Spotify's Top 50 Global daily ranking, which contains the 50 songs with most listeners the day before each edition. To make predictions, the model employs audio features collected provided by the platform API. These features indicate if the songs are dancing, energetic, acoustic, instrumental, among other possibilities.

A previous version of this work was presented at the 2019 edition of the Brazilian Symposium on Computer Music (SBCM) [3]. During this study some models were developed using different approaches, like ranking positions [4] and acoustic characteristics of songs [5]. On the SBCM paper we presented the results of a model that predicts if a song on Spotify's Viral 50 Global ranking will appears on Top 50 Global ranking and vice-versa. The main scope of this paper is different from previous work in that, here, the idea is to predict if a song will be popular even before its release.

The remainder of this paper is organized as follows: In Section 2 we present related work, while in Section 3 we describe our methodology. In Section 4 we show the results we obtained and discuss them in Section 5. Finally, in Section 6 we present our final remarks and point out future directions to be taken.

2. Related Work

HSS models are generally based on supervised machine learning techniques. So, training data is needed to build these models. Different sources of data have been used in the literature. In our studies, we identified three types of data sources as the most common, namely: songs acoustic features, social network information, and concert and festival data. In this section, we present researches that used these data sources to make their predictions.

Regarding the use of concert information to make predictions, Arakelyan et al. [6] collected data from the SongKick website[7]. The data contained the location, list of participating artists, event name and a value indicating the event popularity given by the website. The authors considered an artist to be popular if they had a contract with one of the following record companies: Sony BMG, Universal Music Group or Warner. The labels affiliated with these companies were also considered for success. The authors applied the logistic regression method to predict whether an artist would succeed or not. The maximum accuracy obtained was 39%.

Another work that also used data from concerts and festivals was made by Steininger and Gatzemeier [7]. For each event, the authors obtained some 20 parameters identified with Amazon Mechanical Turk[8] contributors. From this data, they sought to predict whether or not the songs of the artists who participated in these events would appear on a list of Germany's 500 most popular songs in 2011. There is no information from where such list is published. The authors were able to show that there was a correlation between the data with 95% of statistical significance. However, the maximum accuracy obtained was 43.5% using the PLS-SEM approach.

Regarding the use of social network data, Kim, Suh, and Lee [8] collected messages on Twitter associated with the tags: #nowplaying, its abbreviated version (#np), and #itunes (a digital music selling platform). With this data, they sought to predict whether a song would be successful. For the authors, success is achieved when the song appears up to a certain position on the Billboard Hot 100[9] (this position was varied in the experiments). The authors calculated different correlation coefficients between the number of messages collected and the success of each song. The maximum value was 0.41, which may indicate that there is no correlation between them. Even with such an obstacle, the authors applied a random forest classifier, obtaining 90% accuracy in the model where a song is only considered successful if it is in the top ten.

On a different approach, Herremans, Martens, and Sörensen [9] created a model for predicting the popularity of Dance songs using acoustic features. For a song to be considered popular in this research it should be up to a certain position in the Official Charts Company Top 40 Dance Music[10] (just as in the previous work, this position was also varied in the experiments). The authors collected metadata and information from acoustic features of the tracks that appeared in this ranking between 2009 and 2013 using The Echo Nest[11]. Three distinct experiments were performed where different parameters for a song to be considered popular were tested. The best results were obtained by using the Naive Bayes classifier. In such experiment a music should be in the top ten to be considered popular and between positions 31 to 40 to be considered unpopular. Songs in positions 11 to 30 were discarded. Given all these assumptions, the authors obtained accuracy and AUC of 65%.

In addition to this work, Karydis et al. [10] retrieved data associated with 9,193 songs that were featured in at least one popularity ranking from the following sources between April 28th, 2013 and December 28th, 2014: Billboard, Last.fm, and Spotify. Additionally, they retrieved data from 14,192 songs of the albums in which these popular tracks were released. They retrieved this data from three different sources, namely: iTunes[12], Spotify, and 7digital[13]. Plus, using four different tools, they extracted the songs acoustic features from 30-second samples of them. Their goal was to predict which song would be the most successful from an unseen album.

[5] ⟨http://bit.ly/2KwJmGu⟩

[6] ⟨http://bit.ly/2QrRGLs⟩

[7] ⟨https://www.songkick.com/⟩

[8] Amazon Mechanical Turk is a service offered by Amazon for hiring persons to perform tasks virtually. ⟨https://www.mturk.com/⟩

[9] The Billboard Hot 100 is a weekly ranking containing the 100 most popular songs in the United States. ⟨https://www.billboard.com/charts/hot-100⟩

[10] The Official Charts Company publishes rankings of most popular songs, albums and films in the United Kingdom. ⟨http://bit.ly/2FgiuY1⟩

[11] The Echo Nest is the industry's leading music intelligence company, providing developers with the deepest understanding of music content and music fans. ⟨http://the.echonest.com/⟩

[12] ⟨https://apple.co/37qTlWP⟩

[13] ⟨http://docs.7digital.com/⟩

The authors employed two temporal-data models: a nonlinear auto-regressive network classifier (NAR) and its variation with exogenous inputs (NARX). They reported precision of 46% and accuracy of 52%.

Since Pons and Serra [11] showed that neural networks could be a valuable option on HSS researches, Martín-Gutiérrez et al. [12], in a more recent work, used them along with information collected from Spotify and Genius[14] of more than 100 thousand tracks. This data relates to audio features and characteristics, plus knowledge from songs lyrics and files. The authors created a model based on neural networks to predict the song popularity value on Spotify, a number in a scale from 0 to 100. The higher this number, the higher is the song popularity on the platform. The authors obtained an accuracy and recall of 83.46% in the best case when using a neural network with 3 layers and Adam optimizer.

The research that most closely resembles ours is the one by Reiman and Örnell [13]. In this work, the authors collected data from 287 songs that appeared in Billboard Hot 100 between 2016 and 2018. They also collected data from 322 other songs that never appeared in this ranking, randomly chosen from 13 different music genres. The information was collected using the Spotify API and relates to the same audio features we used in our research. As previously stated, these features indicate if the songs are happy, dancing, instrumental, etc. For a song to be considered popular in this research, it should be present in Hot 100.

Reiman and Örnell [13] used four different algorithms to make their predictions, namely: K-Nearest Neighbors, Support Vector Machines, Gaussian Naive Bayes and Logistic Regression. The experimental evaluation was based on hold-out validation (80% for training and 20% for testing) with a maximum accuracy of 60.17%, obtained by Gaussian Naive Bayes. The authors' conclusion is that the experiments have not shown that it is possible to predict whether or not a song will be a success.

3. Methodology

In this section we present the methodology used in this research, beginning with the way the data was collected and prepared (cf. Subsection 3.1). We then present the experiments we carried out (cf. Subsection 3.2), and how we evaluate the results obtained (cf. Subsection 3.3). A graphical representation of the methodology is given in Figure 1.

3.1 Data Collection and Preparation

The data collection was performed using the Spotify Web API[15]. From November 2018 to July 2019 we collected daily information from the Top 50 and Viral 50 public playlists. These playlists act as platform rankings, the first containing the top 50 songs listened the day before, while the second features 50 songs that had the biggest increase in the number of plays the day before[16].

In this work, we consider the songs in the Top 50 to be popular, in an approach already used in other HSS works [9, 13]. Because we require data to be collected from the Spotify API, non-popular songs must still be featured on the platform. Therefore we consider non-popular songs to be those that featured in Viral 50 but did not appear in the Top 50 during the collection period, thus avoiding a song to be simultaneously popular and unpopular.

The data from these rankings was collected using the API's "Get a Playlist's Tracks" function. We retrieved the names of the artists and their tracks, the songs ID's within the platform, and the Explicit flag, which indicates whether the song contains explicit lyrics.

We also collected audio features from each song. To do this, we used their ID's as input to the API's "Get Audio Features for Several Tracks" function. The features used in our experiment are listed below. All features range in $[0, 1]$, with values closer to 1 expressing more strongly the concept of the feature:

- **Danceability:** describes how suitable a track is for dancing, taking into account several factors such as tempo, rythm, and overall regularity;
- **Energy:** represents a "perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy"[17]. Obtained from features such as dynamic range, perceived loudness, timbre, onset rate, and general entropy;
- **Speechiness:** whether the music contains spoken words. According to the official documentation[18], if this measure is above 0.66, then it is probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech—e.g., rap music. Values below 0.33 most likely represent music and other non-speech-like tracks;
- **Acousticness:** gives a confidence level on how acoustic the music is, in terms of relying more on acoustic instruments rather than electronic ones;
- **Instrumentalness:** how prevalent the sound of instrument is rather than vocals. Non-verbal sounds such as "ooh" and "aah" are considered instrumental. According to the documentation, values above 0.5 are represent songs that are mostly instrumental;
- **Liveness:** detects the presence of an audience in the recording. According to the documentation, a value

[14] ⟨https://genius.com/⟩
[15] ⟨https://spoti.fi/37vPA2l⟩
[16] According to Kevin Goldsmith, Spotify's former vice-president of engineering, whose explanation may be found at ⟨http://bit.ly/33fXg67⟩ (requires log in to the platform). Access on 2020-08-13.
[17] ⟨https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/⟩
[18] See footnote 17.

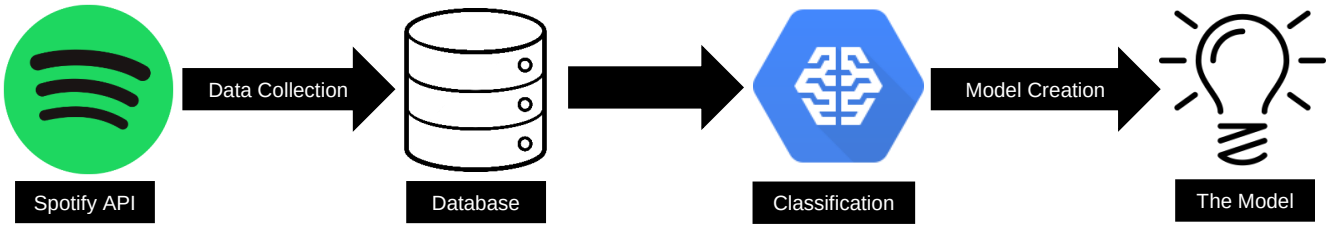

Figure 1. Representation of the methodology used in this work.

above 0.8 indicates with high degree of probability that the song was recorded live;

- **Valence:** the valence is a measure of "positiveness". The higher the valence, the more the song relates to positive feelings, such as happiness and euphoria, whereas low valence resemble negative feelings, such as sadness and anger.

All of these audio features are float fields and the documentation does not tell how they are calculated. Therefore, we cannot compute these values for songs that are not on the platform, making it difficult to make predictions for songs not yet in the platform. To make such predictions viable, we decided to binarize these fields. In the binarization of the collected data, the field was considered positive if its value was greater than 0.5. The exceptions were speechiness and liveness, where we used the values 0.33 and 0.8 as a basis, respectively, due to the description of these fields in the documentation.

In order to make predictions for a song not yet released, even if we do not know the exact value achieved by it in the audio features, the artist themselves can indicate whether or not it is happy, live, dancing, etc. Thus, it is possible to represent unreleased songs as instances of our base, allowing making predictions of its success.

For our experiments, we set up two databases. In the first one, each entry represented one song on a given day, and there might be multiple entries for the same song if it appears more than once in the ranking. In the second, the entries with the same song name and artist were combined into one. In this case a song was only considered popular if it appeared more than a certain number of times in the Top 50 during collection time. After this process, we discard the name fields and the ID's of the two databases.

During the Christmas season it is common for themed songs to appear in the Top 50 from December 23 to 26. To prevent these songs from being taken as popular in the second experiment, we established that for a song to be considered popular it should have appeared more than four times in the Top 50.

For comparison, we set up a model based on the methodology used by Reiman and Örnell [13]. We will use the acronym ROM (Reiman and Örnell Model) when dealing with this model from now on. In this work we used all audio features available in the API except the field "Explicit". Therefore, besides the previously presented features, we also use:

- **Duration_ms:** the duration of the track in milliseconds;
- **Key:** the key the track is in. Integers map to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on;
- **Mode:** indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0;
- **Tempo:** the overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration;
- **Time_signature:** 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).

For the remainder of this paper, we will adopt the acronym PM when dealing with our Proposed Model. In PM, we do not used all audio features available in the API, because we only selected those where it was possible to do the binarization process. In the specific case of the Mode field which is binary, we do not use it because what it represents is directly associated with the musical note, which is represented in the Key field that was discarded since it cannot be binarized.

We remark that the ROM is not an exact reproduction of the methodology used by Reiman and Örnell [13], but a model created based on that text, so modifications were made to fit our experiments. The first difference is in the source of popular and non-popular data. In this work we used Top 50 and Viral 50 as sources of popular and non-popular songs, respectively. On the other hand, Reiman and Örnell used Billboard's Hot 100 as their source for popular works and randomly collected music of different genres from Spotify as non-popular. Also, in that work the audio features were not extracted directly from the Spotify API, as we did. They used the Spotipy[19] Python library. Therefore, there may be differences in the way audio features are calculated in these two cases.

In ROM, as in the base text, we do not perform the binarization process and we do not normalize the data neither. In that paper, it is stated that only the instances where the audio features were in the same range were used. However,

[19] ⟨https://spotipy.readthedocs.io/en/latest/⟩

there is no information on which interval was used, so in our experiments the entire dataset was employed. We also set up two databases for ROM in order to compare against the results obtained with our methodology. The instances of these two databases represent the same entries as the PM ones.

3.2 Experimentation

We used different machine learning algorithms in our experiments. Therefore, it was necessary to divide our databases into training and testing groups. In the first experiment, we used data from November and December 2018 for training. In the second one, the data from January 2019 were also used. Testing has always been performed on the June and July 2019 data. Thus, there is a minimum difference of at least five months between the training and test data dates.

For PM, before training, all the data was standardized by removing the mean and scaling to unit variance. This step was not made for ROM's input as it was not made in the base text. The standard score of a sample x is calculated as $z = (x-u)/s$, where u is the mean of the training samples and s is the standard deviation.

To make the results more comparable, we restricted the number of algorithms used in our experiments to those that were also used by Reiman and Örnell [13]. Thus, the algorithms used were Gaussian Naive Bayes (GNB), K-Nearest Neighbors (KNN), Logistic Regression (LR) and Support Vector Machine (SVM) with RBF kernel. The way these algorithms work will be discussed next.

We used the "scikit-learn" [14] library in our experiments. This library contains implementations of all the algorithms used, as well as being one of the most widely used in academia and market. We used default values in all parameters. Our data were store in .csv files and were accessed using Pandas library [15]. Pandas is one of the most used libraries for data manipulation and analysis in Python programming language [16].

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of the attributes. In the Gaussian Naive Bayes algorithm the likelihood of the features is assumed to be Gaussian [17].

Neighbors-based classification is a type of instance-based learning or non-generalizing learning: it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the nearest neighbors of each point: a query point is assigned the data class which has the most representatives within the nearest neighbors of the point. The scikit-learn's KNeighborsClassifier implements learning based on the K nearest neighbors of each query point, the default value of K is **5** [20].

[20] ⟨http://bit.ly/2Qh3vTY⟩

Logistic regression, despite its name, is a linear model for classification rather than regression. Logistic regression is also known in the literature as logit regression, maximum-entropy classification (MaxEnt) or the log-linear classifier. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a logistic function[21].

SVM is an instance-based classifier that projects the training samples into a space of higher dimensionality, where it assumes to have a representative layout of the original space. In this projection, SVM attempts to find the hyperplane that best separates the classes, effectively dividing the decision space into two subspaces. When classifying a new sample, SVM projects its features into the same high-dimensional space and verifies on which subspace the projected instance "falls", and then assigns it the class label associated with that subspace [18]. The kernel function defines the inner product in the transformed space, so that different kernels imply on different ways to calculate inner products [19].

3.3 Evaluation of Results

To evaluate the results obtained, we use the following metrics, where we denote the number of true positives, true negatives, false positives and false negatives as tp, tn, fp, and fn respectively:

1. **Accuracy =** $\frac{tp+tn}{tp+tn+fp+fn}$, the percentage of correctly predicted instances;
2. **Precision =** $\frac{tp}{tp+fp}$, the percentage of correctly predicted positive instances;
3. **Negative Predictive Value (NPV) =** $\frac{tn}{fn+tn}$, the percentage of correctly predicted negative instances;
4. **Recall =** $\frac{tp}{tp+fn}$, the percentage of true positives;
5. **Specificity =** $\frac{tn}{tn+fp}$, the percentage of true negatives;
6. **F1 Score =** $2\frac{\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}}$, the harmonic mean of precision and recall;
7. **Area Under the Receiver Operating Characteristic Curve (AUC) =** $\int_{\infty}^{-\infty} \text{Recall(T)}\text{Specificity'(T)}\, d\text{T}$, the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance when using normalized units [20];
8. **Matthews Correlation Coefficient (MCC) =** $\frac{tp*tn-fp*fn}{\sqrt{(tp+fp)*(tp+fn)*(tn+fp)*(tn+fn)}}$, a measure of the quality of binary classifications. It takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes [21].

Except when noted, these metrics were defined according to Olson and Delen [22].

[21] ⟨http://bit.ly/2Qgtnj2⟩

4. Results

The confusion matrices obtained in the experiment where the predictions were made on a per-day basis are shown in Tables 1, 2, 3 and 4. A more graphical version of these matrices are also available in the Appendix Section. Table 5 shows the values achieved in the evaluation metrics in this experiment. The best results obtained in each of the metrics are shown in red.

Regarding PM, in this experiment, the best result was obtained, in terms of accuracy, using the SVM classifier. This case has the smallest amount of false positives on the experiment with a value 122% lower than the case using KNN, which has the second lowest amount of these incorrectly predicted instances. However, when using SVM the highest amount of false negatives were also obtained, with a value 34.5% higher than the case using GNB, which presented the lowest amount of these instances.

Due to these factors, the SVM classifier has not obtained the best results in all the metrics used to evaluate the models. However, it has the highest value in MCC, which evaluates the quality of a binary classification. This indicates that the result obtained by this classifier was the best overall.

The confusion matrices obtained in the experiment where the predictions were made on a per-song basis are in the Tables 6, 7, 8 and 9. As before, more graphical versions of these matrices are also available in the Appendix Section. Table 10 presents the values achieved in the evaluation metrics in this experiment. The best results obtained in each of the metrics are shown in red.

Regarding PM, in this experiment, SVM obtained again the highest value in MCC and accuracy, which indicates that it was the one that obtained the best results in general. In this case, the number of false positives and false negatives were the second lowest in comparison to the other models. Thus, unlike the first experiment, the highest F1 Score was also obtained by SVM.

One concern was that the results obtained by ROM would not necessarily represent the results that the original model might obtain. However, the maximum difference obtained in percentage points between the results obtained by ROM in our first experiment and the results presented in the base text was only 6.64 in accuracy when using the GNB classifier.

In that work, the authors stated that it is not possible to make predictions in the music market using audio features. By creating a model based on the methodology proposed in that paper, we could not make good predictions neither. The MCC in the experiments did not exceed 0.14 in neither case, indicating an unsatisfactory binary classification. However, our proposed methodology allowed predictions with MCC greater than 0.7. This result indicates that is possible to predict if songs will be popular, even before their releases, using audio features.

5. Discussion

In this research, we developed a machine learning-based model to predict whether or not an unreleased song will become popular. Specifically, we predict whether or not a song will appear in Spotify's Top 50 ranking. However, we remark that our methodology could be reproduced on any streaming platform, provided the audio features and the binary accoustic features are available. We performed two experiments. In the first, each instance represented one song on a specific day of the rankings collected (Spotify's Top 50 and Viral 50), so there were several identical instances that represented the same songs. In the second, the instances that represented the same song were merged into one entry. Thus, in the first experiment the algorithms were trained with 5389 instances and in the second with 405.

In the first experiment, the predictions were made for individual ranking editions. That is, a song was considered popular on a specific day if it appeared in the Top 50 of that day. In contrast, in the second experiment the predictions were made for a set of rankings. In this case, for a song to be considered popular, it should appear a certain number of times in the Top 50. We decided to set this value on four appearances, as this value prevents songs that stood out only from December 23 to 26 to be considered as popular.

Despite the discrepancy in the number of training instances in the experiments, MP obtained similar results in both of them, showing that it can achieve a good learning even with a small amount of data. The SVM classifier with RBF kernel obtained the highest values in MCC, AUC and accuracy in our experiments. Comparing the results in the two cases, the difference in accuracy was 5.7 percentage points, while it was 0.23 in AUC and 5.17 in MCC. The second experiment obtained the highest values in these metrics.

The results obtained by MP differ from those obtained by ROM. MP presented, in the best case of both models, 56.65% higher accuracy in the first experiment and MCC 921.02% higher in the second. In Table 11 we show the percentage of superior performance of MP compared to ROM in the two experiments we performed. For this calculation we used the values of the best models for each test – SVM for MP and KNN for ROM.

One possible explanation for the poor result obtained by ROM is that there is no data preparation in this methodology. The authors did not normalize the attributes used in their research, thus hindering the learning of their models, because they are exposed to atypical values and with great variation. On the other hand, in our model, in addition to not utilizing the full set of information available through the Spotify API, we transform the attributes into binary fields. This process removes the need for normalization, facilitates learning, and even allows us to make predictions for unreleased songs.

One study [23] has already shown that popular songs tend to sound similar. This study analysed 500,000 albums from 15 different genres. The authors evaluated the complexity of each song, calculated from the tracks acoustic features,

Table 1. Confusion matrices of the experiment where the predictions were made on a per-day basis using SVM classifier.

		Predicted Label			
		PM		ROM	
		0	1	0	1
True Label	0	2129	85	1280	934
	1	697	2342	1519	1520

Table 2. Confusion matrices of the experiment where the predictions were made on a per-day basis using Gaussian Naive Bayes classifier.

		Predicted Label			
		PM		ROM	
		0	1	0	1
True Label	0	1828	386	744	1470
	1	412	2627	971	2068

Table 3. Confusion matrices of the experiment where the predictions were on a per-day basis using Logistic Regression.

		Predicted Label			
		PM		ROM	
		0	1	0	1
True Label	0	1841	373	761	1453
	1	466	2573	997	2042

Table 4. Confusion matrices of the experiment where the predictions were made on a per-day basis using KNN.

		Predicted Label			
		PM		ROM	
		0	1	0	1
True Label	0	1921	293	1297	917
	1	550	2489	1482	1557

Table 5. Performance of the models for the experiment where the predictions were made by day.

	SVM		GNB		LR		KNN	
	PM	ROM	PM	ROM	PM	ROM	PM	ROM
Accuracy	0.8511	0.5330	0.8481	0.5353	0.8403	0.5336	0.8395	0.5433
Precision	0.9650	0.6194	0.8719	0.5845	0.8734	0.5843	0.8947	0.6293
NPV	0.7534	0.4573	0.8161	0.4338	0.7980	0.4329	0.7774	0.4667
Recall	0.7706	0.5002	0.8644	0.6805	0.8467	0.6719	0.8190	0.5123
Specificity	0.9616	0.5781	0.8257	0.3360	0.8315	0.3437	0.8677	0.5858
F1 Score	0.8569	0.5534	0.8681	0.6289	0.8598	0.6250	0.8552	0.5648
AUC	0.8661	0.5391	0.8450	0.5083	0.8391	0.5078	0.8433	0.5491
MCC	0.7253	0.0775	0.6890	0.0174	0.6748	0.0164	0.6793	0.0971

Table 6. Confusion matrices of the experiment where the predictions were made on a per-song basis using SVM classifier.

		Predicted Label			
		PM		ROM	
		0	1	0	1
True Label	0	184	6	117	73
	1	19	63	39	43

and compared these values to the amount of sales of these albums. By applying Pearson's correlation coefficient to the data, the authors obtained a value of -0.69 with p-value equal to 0.001, which demonstrates the statistical significance of this result. Thus, they demonstrated that there is a negative linear correlation between the data. This result indicates that the more complex a song is, the less it tends to get higher sales. This situation may explain how our proposed model has achieved such good results, as it learns some characteristics associated with popular musics.

Table 7. Confusion matrices of the experiment where the predictions were made on a per-song basis using Gaussian Naive Bayes classifier.

		Predicted Label			
		PM		ROM	
		0	1	0	1
True Label	0	168	22	156	34
	1	20	62	66	16

Table 8. Confusion matrices of the experiment where the predictions were made on a per-song basis using Logistic Regression.

		Predicted Label			
		PM		ROM	
		0	1	0	1
True Label	0	156	34	182	8
	1	14	68	78	4

Table 9. Confusion matrices of the experiment where the predictions were made on a per-song basis using KNN.

		Predicted Label			
		PM		ROM	
		0	1	0	1
True Label	0	186	4	150	40
	1	31	51	59	23

Table 10. Performance of the models for the experiment where the predictions were made per song.

	SVM		GNB		LR		KNN	
	PM	ROM	PM	ROM	PM	ROM	PM	ROM
Accuracy	0.9081	0.5882	0.8456	0.6324	0.8235	0.6838	0.8713	0.6360
Precision	0.9130	0.3707	0.7381	0.3200	0.6667	0.3333	0.9273	0.3651
NPV	0.9064	0.7500	0.8936	0.7027	0.9176	0.7000	0.8571	0.7177
Recall	0.7683	0.5244	0.7561	0.1951	0.8293	0.0488	0.6220	0.2805
Specificity	0.9684	0.6158	0.8842	0.8211	0.8211	0.9579	0.9789	0.7895
F1 Score	0.8344	0.4343	0.7470	0.2424	0.7391	0.0851	0.7445	0.3172
AUC	0.8684	0.5701	0.8603	0.5081	0.8560	0.5033	0.8004	0.5350
MCC	0.7770	0.1301	0.6360	0.0192	0.6164	0.0149	0.6866	0.0761

Table 11. Higher performance percentages achieved by PM over ROM.

	Experiment 1	Experiment 2
Accuracy	56.65%	42.78%
Precision	53.34%	150.07%
NPV	61.43%	26.29%
Recall	50.42%	173.90%
Specificity	64.15%	22.66%
F1 Score	51.72%	163.05%
AUC	57.73%	63.32%
MCC	646.96%	921.02%

6. Conclusion and Future Work

In this paper, we presented a model for predicting whether a particular song will be popular on Spotify, one of today's largest music streaming platforms. For a song to be considered popular in this research it must appear in the Top 50 Global ranking, which features Spotify's 50 most popular songs.

To create our model, we set up a database containing songs that had already appeared in the Top 50 and others that were never there. Using the platform's own API we extract information about the database songs. The information collected indicates if the songs are dancing, acoustic, instrumental, etc. This data is collected in float numbers. To allow the inclusion of songs not yet released, we decided to binarize these attributes. This way, the artist or label can determine whether or not their music has these characteristics without having to make use of the API.

Alongside our proposed model, we also developed another one based on the methodology used by Reiman and Örnell [13] in order to compare the results obtained.

We performed two experiments. In the first one, predictions were made on a per-day basis, that is, we sought to predict which songs would be popular on a specific day. So, a song that only appeared in the Top 50 once was considered popular on that particular day. Therefore, this experiment relied on repeated instances with possible distinct classes. On the other hand, in the second one, the predictions were made on a per-song basis, so each instance represented a distinct

song and it was only considered popular if it had appeared at least four times in the Top 50. Despite this, the model obtained similar results in both experiments with a maximum difference of 5.7 percentage points in accuracy.

The proposed model obtained accuracy, precision and AUC above 80% in all cases. In the best case, using the SVM classifier with RBF kernel, the result was more than 920% higher, according to the Matthews Correlation Coefficient, than the Reiman and Örnell [13] based model.

However, improvements can still be done, since the number of false negatives obtained by the proposed model is still high, after all about 23% of positive instances were predicted erroneously. We believe that a possible way to reverse this situation is to add information from social networks to our model. This belief is given because a research [24] has shown that there is a linear correlation between the popularity of an album on Spotify and the amount of positively polarized messages about the artist of this album on Twitter.

Plus, we didn't consider in this work the impact of the artist previous popularity and marketing investing by them or their record labels to boost their songs popularity. Our idea is to also use these information on a future model to reach out better results.

In addition, we also intend to partner with record labels and artists to apply the proposed model to songs before they are released. In our experiment, because we do not have access to these songs, the tests were made considering songs already released as if they were unreleased songs.

Finally, we highlight that although this work was developed focusing on Spotify, its methodology can be easily replicated to other platforms that contain music rankings. Moreover, it would easily be possible to experiment with other success parameters, which may be necessary, because artists at different levels of fame may have different parameters.

Acknowledgements

The results here presented were reached during the master's of Carlos V. S. Araujo which was fund by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

Author contributions

The work was developed by Msc. Carlos V. S. Araujo with orientation of Prof. Rafael Giusti and co-orientation of Prof. Marco A. P. Cristo.

References

[1] PACHET, F.; SONY, C. Hit song science. *Music data mining*, Chapman & Hall/CRC Press Boca Raton, FL, p. 305–326, 2011.

[2] LI, T.; OGIHARA, M.; TZANETAKIS, G. *Music data mining*. [S.l.]: CRC Press, 2011.

[3] ARAUJO, C.; CRISTO, M.; GIUSTI, R. Predicting music popularity on streaming platforms. In: *Anais do XVII Simpósio Brasileiro de Computação Musical*. Porto Alegre, RS, Brasil: SBC, 2019. p. 141–148. Disponível em: ⟨https://sol.sbc.org.br/index.php/sbcm/article/view/10436⟩.

[4] ARAUJO, C.; CRISTO, M.; GIUSTI, R. Will I Remain Popular? A Study Case on Spotify. In: *Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional*. Porto Alegre, RS, Brasil: SBC, 2019. p. 599–610. Disponível em: ⟨https://sol.sbc.org.br/index.php/eniac/article/view/9318⟩.

[5] ARAUJO, C. V. S.; CRISTO, M. A. P. de; GIUSTI, R. Predicting music popularity using music charts. In: *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. [S.l.: s.n.], 2019. p. 859–864.

[6] ARAKELYAN, S. et al. Mining and forecasting career trajectories of music artists. *CoRR*, abs/1805.03324, 2018. Disponível em: ⟨http://arxiv.org/abs/1805.03324⟩.

[7] STEININGER, D. M.; GATZEMEIER, S. Using the wisdom of the crowd to predict popular music chart success. In: *ECIS*. [S.l.: s.n.], 2013. p. 215.

[8] KIM, Y.; SUH, B.; LEE, K. #Nowplaying the Future Billboard: Mining Music Listening Behaviors of Twitter Users for Hit Song Prediction. In: *Proceedings of the First International Workshop on Social Media Retrieval and Analysis*. New York, NY, USA: ACM, 2014. (SoMeRA '14), p. 51–56. Disponível em: ⟨http://doi.acm.org/10.1145/2632188.2632206⟩.

[9] HERREMANS, D.; MARTENS, D.; SÖRENSEN, K. Dance hit song prediction. *Journal of New Music Research*, Routledge, v. 43, n. 3, p. 291–302, 2014. Disponível em: ⟨https://doi.org/10.1080/09298215.2014.881888⟩.

[10] KARYDIS, I. et al. Musical track popularity mining dataset: Extension & experimentation. *Neurocomputing*, v. 280, p. 76 – 85, 2018. Applications of Neural Modeling in the new era for data and IT. Disponível em: ⟨http://www.sciencedirect.com/science/article/pii/S0925231217317666⟩.

[11] PONS, J.; SERRA, X. Randomly weighted cnns for (music) audio classification. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.: s.n.], 2019. p. 336–340.

[12] MARTÍN-GUTIÉRREZ, D. et al. A multimodal end-to-end deep learning architecture for music popularity prediction. *IEEE Access*, v. 8, p. 39361–39374, 2020.

[13] REIMAN, M.; ÖRNELL, P. Predicting hit songs with machine learning. In: . [S.l.: s.n.], 2018. (TRITA-EECS-EX, 2018:202).

[14] PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

[15] REBACK, J. et al. *pandas-dev/pandas: Pandas 1.1.3*. Zenodo, 2020. Disponível em: ⟨https://doi.org/10.5281/zenodo.4067057⟩.

[16] ROSSUM, G. V.; JR, F. L. D. *Python tutorial*. [S.l.]: Centrum voor Wiskunde en Informatica Amsterdam, 1995. v. 620.

[17] ZHANG, H. The optimality of naive bayes. *AA*, v. 1, n. 2, p. 3, 2004.

[18] SCHÖLKOPF, B. et al. Estimating the support of a high-dimensional distribution. *Neural Computation*, v. 13, n. 7, p. 1443–1471, 2001. Disponível em: ⟨https://doi.org/10.1162/089976601750264965⟩.

[19] HERBRICH, R. *Learning kernel classifiers: theory and algorithms*. Massachusetts: MIT press, 2001.

[20] FAWCETT, T. An introduction to roc analysis. *Pattern Recognition Letters*, v. 27, n. 8, p. 861 – 874, 2006. ROC Analysis in Pattern Recognition. Disponível em: ⟨http://www.sciencedirect.com/science/article/pii/S016786550500303X⟩.

[21] BOUGHORBEL, S.; JARRAY, F.; EL-ANBARI, M. Optimal classifier for imbalanced data using matthews correlation coefficient metric. *PLOS ONE*, Public Library of Science, v. 12, n. 6, p. 1–17, 06 2017. Disponível em: ⟨https://doi.org/10.1371/journal.pone.0177678⟩.

[22] OLSON, D. L.; DELEN, D. *Advanced data mining techniques*. [S.l.]: Springer Science & Business Media, 2008.

[23] PERCINO, G.; KLIMEK, P.; THURNER, S. Instrumentational complexity of music genres and why simplicity sells. *PLOS ONE*, Public Library of Science, v. 9, n. 12, p. 1–16, 12 2015. Disponível em: ⟨https://doi.org/10.1371/journal.pone.0115255⟩.

[24] ARAUJO, C. V. et al. Predicting music success based on users' comments on online social networks. In: *Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web*. New York, NY, USA: ACM, 2017. (WebMedia '17), p. 149–156. Disponível em: ⟨http://doi.acm.org/10.1145/3126858.3126885⟩.