Predicting Music Popularity on Streaming Platforms

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

[1]Federal University of Amazonas
Manaus, Brazil

{vicente, marco.cristo, rgiusti}@icomp.ufam.edu.br

Abstract. *Online streaming platforms have become one of the most important forms of music consumption. Most streaming platforms provide tools to assess the popularity of a song in the forms of scores and rankings. In this paper, we address two issues related to song popularity. First, we predict whether an already popular song may attract higher-than-average public interest and become "viral". Second, we predict whether sudden spikes in public interest will translate into long-term popularity growth. We base our findings in data from the streaming platform Spotify and consider appearances in its "Most-Popular" list as indicative of popularity, and appearances in its "Virals" list as indicative of interest growth. We approach the problem as a classification task and employ a Support Vector Machine model built on popularity information to predict interest, and vice versa. We also verify if acoustic information can provide useful features for both tasks. Our results show that the popularity information alone is sufficient to predict future interest growth, achieving a F1-score above 90% at predicting whether a song will be featured in the "Virals" list after being observed in the "Most-Popular".*

1 Introduction

The global entertainment market (movies, games, music, and television) is a billion-dollar industry. According to the Recording Industry Association of America – RIAA –, in 2018 the music industry was worth US$ 9.8 billion in the United States alone, 75% of which were due to streaming services and 11% to downloadable media[1]. Also according to the International Federation of the Phonographic Industry – IFPI –, in the same year the global industry was worth US$ 17.3 billion, 38% of which were due to streaming services and 16% to downloadable media[2].

It is no surprise that such market is fiercely competitive. Wikipedia lists over 1,400 record labels in the United States alone[3]. In face of such competition, the understanding of what makes an album or song successful is key information. It could be used to plan better marketing campaigns, to decide the best moment for the release of a new album or song, and to align the artists' effort with public interest in, e.g., genre, theme, etc.

The success of an album or single song may be assessed in several manners. The most common are probably by means of rankings, such as those provided by the American magazine Billboard, who has been evaluating the music market since the 1940s. Some of the most famous rankings from this magazine are the Hot 100 Songs and the Billboard 200[4]. The Billboard Hot 100 Songs list ranks the most popular songs from each week, while the latter ranks the most popular albums. Billboard also provides rankings that are specific to genres, countries, and distribution methods, as well as year-end rankings, which are versions of the previously mentioned charts, but for the popularity of music and albums over the entire year [1]. In spite of being fairly known to the general public, and also having been used for prediction of popularity in the past [2], Billboard charts are specific to the American market, and we do not employ them as data source in our work.

To assess the global market, we must focus on platforms that provide worldwide service. According to the IFPI, music streaming has become the most popular method of music distribution[2], finally surpassing physical media in 2017. Streaming services also provide several types of statistics concerning artists, albums, and scores, which we may explore to measure worldwide popularity. We shall focus our attention on the streaming platform Spotify, but we do note that our results may be extended to incorporate data from other streaming platforms, such as Tidal and Apple Music.

The problem of predicting success in the musical market has been addressed in the literature before, with Machine Learning algorithms providing the best results in general. Some authors employed acoustic features to create predictive models [3, 4], while others resorted to social network data [5, 6]. Our proposal relies only on historical data of popularity, which we measure with the aid of streaming services, to predict the continued success or popularity growth of a song. The major reason for seeking a model that is free from acoustic data is that such data may not always be available—Spotify, for instance, only provides a 30-second sample that may not be the most representative segment of a song. While social network data may be very rich, its collection and preprocessing may be very labour-intensive and expensive.

The remainder of this paper is as follows. In section 2 we discuss related work, mostly focusing on previous works that predicts popularity from music charts and social network data. Spotify's "Most-Popular" and "Virals" lists are presented in Section 3. The methodology used at this research is described in section 4, and our results are presented in Section 5. Finally, in Section 6 we draw our conclusions and discuss possibilities for future

*Supported by CAPES.

[1] www.riaa.com/wp-content/uploads/2019/02/RIAA-2018-Year-End-Music-Industry-Revenue-Report.pdf

[2] www.ifpi.org/downloads/GMR2018.pdf

[3] en.wikipedia.org/wiki/Category:American_record_labels. Visited 2019-06-10.

[4] https://www.billboard.com/charts

work.

2 Related Work

We have observed few popular approaches to the problem of music success prediction, despite it having received growing attention for several years. We remark two general strategies: the first uses social network data to assess current public perception and extrapolate how successful a song or album will be in the future. The second relies on acoustic information of previously successful songs to predict the success of a song or album.

The work by Kim et al. [7] is an example of the first strategy. The authors collected their data from the social network Twitter. Specifically, they analyzed posts (tweets) associated with the tags *nowplaying*, its shortened version *np*, and *itunes* (Apple's digital music sale platform). The task at hand was to predict whether a music would be successful, which the authors defined as being featured among the 10 first entries in the Billboard Hot-100 ranking. Despite having found a correlation of only 0.41 between the number of tweets and the number of weeks a song stayed in the ranking, the authors observed that a Random Forest classifier was able to achieve an accuracy of 0.90.

Araujo et al. [8] also made use of Twitter data. The authors collected tweets with mentions to 15 popular albums from 2016 and 2017. Their goal was to predict the number of Billboard units achieved by the albums and also their popularity according to Spotify. A Billboard unit can be reached with a single physical or digital sale or with 1,500 music streams. Sentiment analysis was employed to verify whether tweet mentions were positive or negative. The authors observed linear correlation between positive tweets and Spotify popularity, but no correlation between negative tweets and Spotify popularity, neither between tweets of any polarity and Billboard data. The authors hypothesized that Billboard's choice of prioritizing physical sales does not reflect the modern market.

Among the works that resorted to acoustic data, Lee and Lee [3] observed 867 songs that made it to at least three consecutive weeks in Billboard's "Hot Rock Songs", a weekly ranking that features 50 entries of rock music. For each song, the authors collect a 30-second sample and extracted acoustic information, such as chroma, rhythm, and timbre. In addition to the acoustic data, they also employed information on whether the artist had been previously featured in the ranking. Those data were used to train a multi-layer perceptron classifier [9], and the authors' task was to predict how many weeks a song would remain in the ranking. Their model achieved an accuracy of 0.55 when only acoustic information was used and 0.54 when the model was trained with only information about previous appearances of the artist in the ranking. When both types of data were combined, the model achieved an accuracy of 0.59.

Karydis et al. [4] 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 songs of the albums in which these popular tracks were released. This resulted in a data set of popularity scores and acoustic information of 23,385 songs. Their goal was to employ knowledge of the most successful songs from past albums to predict which song will be the most successful from an unseen album. The authors employed two temporal-data models, namely a non-linear autoregressive network classifier (NAR) and its variation with exogenous inputs (NARX). The authors reported precision of 0.46 and accuracy of 0.52.

In addition to the previously discussed strategies, Arakelyan et al. [10] compiled data from SongKick[5] about live performances and festivals. Their task was to predict whether artists feature in those performances and festivals would sign contracts with major record labels. They employed logistic regressors and reported precision of 0.39. Steininger and Gatzemeier [11] analyzed data from live performances in Germany and their task was to predict whether songs from the observed artists would be featured among the 500 most popular German songs in 2011. Using Partial Least Squares Structural Equation Modeling (PLS-SEM), the authors reported an estimated precision of 0.43.

3 Spotify's Lists

We collected our data from the streaming platform Spotify. Spotify is the third largest music streaming platform, according to Forbes[6]. Spotify publishes daily lists of popular and "viral" songs which, according to Kevin Goldsmith[7], a company's former vice-president of engineering, are constructed in the following ways: the "Most-Popular" list ranks songs according to the total number of streams in the previous day, while the "Virals" list ranks song according to the growth in number of streams.

From Goldsmith's description we draw that a song will remain among the "Virals" if its number of listeners is constantly rising, which implies that the song is reaching a broader public than what is usual for a particular artist. It seems reasonable to assume that virality is an event desired by artists who want to expand their audience. However this definition of virality also means that already successful artists will find it more challenging to hit the "Virals" list than the "Most-Popular" list. On the other hand, less famous artists will tend to find it harder to reach the "Most-Popular" list, being more likely to be featured among the "Virals".

Our goal is to predict whether an already-popular artist may experience sudden growth in public interest. We

[5] https://www.songkick.com/

[6] https://www.forbes.com/sites/hughmcintyre/2018/05/25/the-top-10-streaming-music-services-by-number-of-users. Visited 2019-06-11.

[7] https://www.quora.com/What-are-Spotifys-US-Viral-50-and-Global-Viral-50-charts-Are-these-just-daily-charts. The answer is visible only to logged in users. Visited at 2019-06-11.

do so by making use of data from the "Most-Popular" list to predict appearances in the "Virals". The converse may also be of great interest. Once a piece of work reaches the status of viral, is the sudden spike in popularity merely incidental and temporary? Or does the regular audience remains more expressive in the long term? To answer this, we predict whether a music that is featured in the "Virals" list will be featured in the "Most-Popular".

4 Methodology

In this section we present our experimental method. We propose four models for predicting appearances in Spotify's "Virals" list from appearances in its "Most-Popular" list, and vice versa. The first employs previous data from one list to make predictions in the other, and the second extends this model with acoustic information. The third model makes predictions using acoustic information only. And the fourth model is a baseline, which only counts appearances in one list and predicts appearances in the other if the song has surpassed a threshold in the first list.

4.1 The Classification Problem

Our experimental method is outlined in Figure 1. It consists of the following steps: data collection, extraction of acoustic features, baseline generation, data set preparation, models training and testing, and analysis of results.

Before we discuss how we collected the data and constructed the models, we shall define our classification problem, which we divide in two phases.

During the *assessment phase*, we collected data for days $D_1, D_2, \ldots, D_n$. We shall discuss the data in Section 4.2 but, for the moment, it suffices to say that the data for each day D_i is represented as a pair of lists (P_i, V_i) that contains information for the 50 "Most-Popular" songs and the 50 "Virals" songs of that day. The parameter n is the number of days for which we collected training data.

During the *prediction phase*, we aim to answer the following questions:

1. For every song featured in the "Most-Popular" list $P_{t>n+k}$, will it also be featured in the "Virals" list V_{t+1}?
2. For every song featured in the "Virals" list $V_{t>n+k}$, will it also be featured in the "Most-Popular" list P_{t+1}?

In both cases, t and $t+1$ are the days for which we want to make predictions. The inequality $t > n+k$ simply means that the assessment phase and the prediction phase do not have overlapping days, and that the prediction phase starts k days after the assessment phase has ended. This gap is imposed to avoid overlapping information from the lagged features, discussed in section 4.4, therefore k is a hyperparameter in our models.

The models themselves are discussed in depth in Section 4.5. At this time, we want to make the reader aware that the models were trained with data from the assessment phase only. This restriction allows us to predict several days after the assessment phase, allowing for early prediction of popularity and "viralization" phenomena.

4.2 Data Collection

Data from the "Virals" and "Most-Popular" lists were collected using Spotify's Web API[8]. The data were collected on a daily basis between November, 2018 and January, 2019.

For each daily list, we collected information for nine fields made available by the API. Namely, the rank and the date of the ranking, the names of the artists and of the song, date of release of the song, duration in milliseconds and a URL for a 30-second sample of the song. We note that the URL was not available for roughly 3% of the songs feature in our data, so we removed any rows where this field was empty. Additionally, each song has an "explicit" flag, which indicates whether it contains profanity, and a popularity score, which is a value in the $[0, 100]$ interval that reflects how popular the song is.

Finally, we downloaded the 30-second sample of each song and extracted five acoustic features. The following features were extracted with the Python package LibROSA [12]:

1. Mel-Frequency Cepstral Coefficients (MFCC): obtained from the cepstrum of the compressed mel representation of the signal. The MFCC is probably one of the most often used features in speech processing, and is an expressive low-dimensional representation of a signal. In this work we used 13 coefficients per song;
2. Spectral Centroid: the centroid of each frame of a magnitude spectrogram that has been normalized and treated as a distribution over frequency bins;
3. Spectral Flatness: a measure of noise-like a sound is, as opposed to being tone-like [13];
4. Zero Crossings: the number of times a waveform changes sign;
5. Tempo: number of beats per minute.

4.3 The Baseline

The baseline is a low-effort approach to answer questions 1 and 2 using the least amount of available information. It should be straightforward and, therefore, easily surpassed by a model specifically designed to make popularity predictions. The process of constructing the baseline is closely related to how we make the data set, so we shall present it before the actual models.

During the assessment phase, to each individual song is assigned a "Popularity-Presence" score and a "Viral-Presence" score, which is the number of times that a particular song was featured in a list. For instance, if song x_j was featured in the "Most-Popular" list for two consecutive weeks in November, and then for 5 non-consecutive days in December, then its "Popularity-Presence" score

[8] `https://developer.spotify.com/documentation/web-api/`

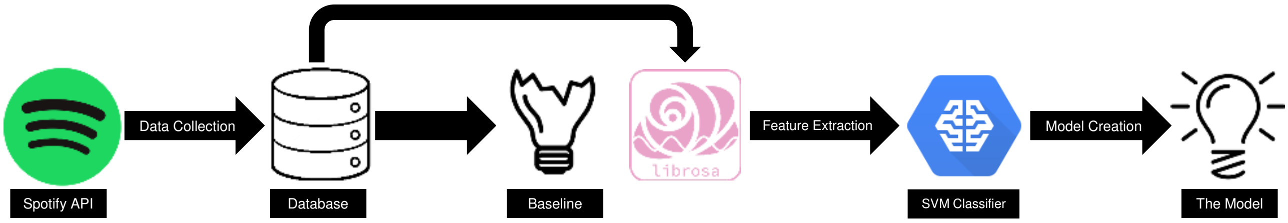

Figure 1: Outline of our experimental method.

will be $p_j = 19$. Similarly, if the same song appears in 13 daily "Viral" lists, then its "Viral-Presence" score will be $p_v = 13$.

These scores are used to define two thresholds that will be used by the baseline model. The "Most-Popular" threshold θ_p is defined as the the lowest value of "Popularity-Presence" score a song must have to be considered probable to be featured again in that list. We empirically set θ_p to be the median of all "Popularity-Presence" scores after looking at the distribution of the scores. Similarly, we defined the "Viral" threshold θ_v as median of the "Viral-Presence" scores.

As explained in Section 4.1, the classification problem we are tackling requires predicting whether a song will appear in one list after being observed in the other. Therefore, we define the popularity baseline as follows. During the prediction phase, for some $t > n$, if a song x_j appears in the "Most-Popular" list P_t and p_j is at least as large as the popularity-baseline threshold, then it will appear in the "Viral" list V_{t+1}. In other words, a song featured in today's "Most-Popular" will be featured in tomorrow's "Viral" if it was featured more than the popularity-baseline threshold during the assessment phase.

Similarly, the viral baseline will predict that a song x_j that appears in the "Viral" list V_t will be featured in the "Most-Popular" list P_{t+1} if v_j is at least as large as the viral-baseline threshold.

4.4 Data Set Creation

The data set was created from the lists collected during the whole period in a number of steps. The first step was the insertion of cross-information between the two types of lists. That is, for entries from the "Most-Popular" list we added relevant "Viral" information and vice versa. The cross-information for each list P_i and V_i varies daily, and includes the position of a song in the other list, the number of consecutive days the song has been featured in the other list. Because the classification task is to predict whether a song that appears in one list will also appear in the other list on the next day, we add to each "Virals" entry a flag indicating whether it is featured in the "Most-Popular" list on the next day and vice versa. Specifically, that flag is the class label. Furthermore, if a song has already been featured in the other list at least once prior to the i-th day, then we also add the date in which it first appeared.

With exception of the 30-second sample URL, no attributes have missing values when collected through the API. However, the insertion of cross-information may cause missing values to appear. This could happen, for instance, if a song from the "Virals" list has never appeared in the "Most-Popular" in the assessment phase. Because of the nature of those data, we simply replace missing data with zeros.

Finally, all non-numeric fields were transformed into categories through one-hot encoding [14]. And because the URLs to the 30-second samples were only required to extract acoustic features, they were removed from the data set at this point.

To capture temporal patterns present in the data, we add lagged features to each instance [15]. In particular, we lag the rank of each song in each list. For instance, at day D_3 in the "Most-Popular" list, an entry $s_{3,i}$ for some song x_j has the features $r_{3,i}$, $r_{2,i}$, and $r_{1,i}$, which express the position of x_j in the "Most-Popular" list during days D_3, D_2, and D_1. Similarly, we lag the rank of each song in the "Virals" list.

4.5 Data Prediction

Before instantiating any models, we first partitioned the data into training and test sets. The data of the lists collected between November 1st and December 31st of 2018 were used to make the training set. There were several test sets, which reflects from our experimental setting of evaluating the models for short-term prediction, mid-term prediction, and long-term prediction. The data used to build the test sets were collected between January 3rd and January 30th of 2019. The two first days of January were not included neither in the training nor in the test because lagged features cause information overlap with data from the training set.

For each list, we produced four test sets. The first set, which we shall refer to as 1st Week, contains data from the lists collected between January 3rd through January 9th. The second set, namely 2nd Week, contains data from the lists collected between January 10th through January 16th. Similarly, the next two sets, 3rd Week and 4th Week, contain data collected until January 23rd and 30th, respectively. Each test set has approximately 350 entries for each list—fewer than that if any entry was removed due to not containing a URL to its 30-second sample.

Recall that, for each day D_t, the class is whether the song will be featured in the other list on day D_{t+1}. Therefore the instances associated with the first day of the 1st Week in the "Virals" prediction, for example, are songs from the “Most-Popular” list from Jan. 3. Notice that we can only make “Virals” claims for songs that appear in the “Most-Popular” list the previous day. Therefore, a true positive will be a song from the “Most-Popular” that we claimed to appear in the “Virals” list the next day and indeed appeared in the “Virals” list the next day, while a true negative will be a song that appears in the “Most-Popular” in one day but we claimed to *not* appear in the “Virals” the next day—and verify to have predicted correctly. Similarly for “Most-Popular” predictions we can only make claims for songs that were featured in the “Virals” list the day before.

To make this prediction, we trained a Support Vector Machine (SVM) classifier [16]. The 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, the SVM attempts to find the hyperplane that best separates the classes, effectively dividing the decision space into two subspaces. When classifying a new sample, the 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 [17].

The model was trained and tested with the library “scikit-learn” [18], which implements its own version of the SVM inducer based on the LibSVM implementation [19]. We used the *RBF* kernel and kept recommended default values for all parameters.

The results obtained were evaluated using seven distinct metrics. Namely, accuracy, Area Under the ROC Curve (AUC), Matthews Correlation Coefficient (MCC), sensitivity, F1 Score, precision and specificity.

5 Results

In total, we collected 92 instances of the “Most-Popular” and “Viral” lists from Spotify, resulting in 9,200 rows of data. We removed 132 rows from the “Virals” and 200 rows from the “Most-Popular” lists that were lacking the URL for the 30-second sample. The remaining 8,868 rows contain data associated with 400 unique songs, out of which 231 are featured only in the “Virals” list and 116 are featured in the “Most-Popular” lists. Only 53 songs appeared in both lists.

The confusion matrices for each test round are available at `http://bit.ly/sbcmpaper`. In Figure 2 we show the confusion matrix of our proposal for the fourth week of “Most-Popular” prediction. This matrix represents the round that gave the highest AUC score, with only six misclassified instances. It is also noticeable that the number of negative instances is nearly twice the number of positive instances. This is also observed in the “Virals” list, as shown in Figure 3. We note that this is the worst result achieved by the acoustic features-based model. This experiment also refers to the fourth week of the prediction phase.

The results for the baseline are given in Table 1 and the results for the model that uses only the lists information are given in Table 2. The results for the model trained only with acoustic features are given in Table 3 and the results for the model that makes use of all types of data are given in Table 4.

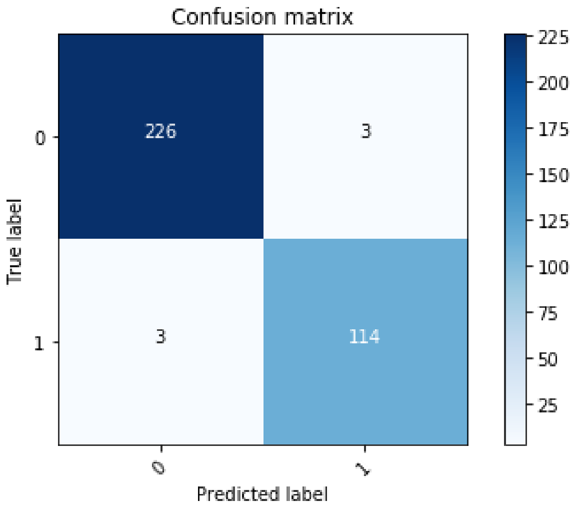

Figure 2: Confusion matrix of the proposed model with only previous list data for predicting the 4th week of the “Most-Popular” list.

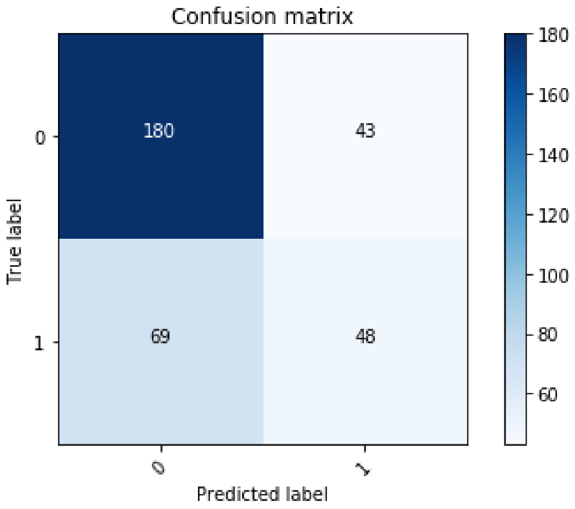

Figure 3: Confusion matrix of the acoustic feature-based model for predicting the 4th week of the “Virals” list.

Initially, we note that the two baselines showed very different behaviors. The “Virals” baseline has MCC values much closer to 0 than 1, which indicates low predictive power. Furthermore, the precision is below 0.5 in the fourth week. This is expected, of course, considering this is the baseline.

On the other hand, the baseline is surprisingly effective in predicting “Most-Popular” songs. The “Most-Popular” list is much less volatile than the “Virals”, which featured 68% more individual songs than the former for the same period. We however note that the performance of this

	Viral				Top			
	1st Week	**2nd Week**	**3rd Week**	**4th week**	**1st Week**	**2nd Week**	**3rd Week**	**4th week**
Accuracy	0.7062	0.7560	0.6932	0.6416	0.9475	0.9296	0.8675	0.8526
AUC	0.6970	0.7465	0.6691	0.6217	0.9182	0.8856	0.8070	0.7862
MCC	0.3715	0.4662	0.3293	0.2196	0.8746	0.8247	0.7148	0.6741
Sensitivity	0.6731	0.7228	0.5982	0.5214	0.8447	0.7835	0.6140	0.5812
F1 Score	0.5858	0.6404	0.5630	0.4959	0.9063	0.8636	0.7609	0.7273
Precision	0.5185	0.5748	0.5317	0.4729	0.9775	0.9620	1.0000	0.9714
Specificity	0.7210	0.7702	0.7401	0.7031	0.9917	0.9877	1.0000	0.9913

Table 1: Performance of the baseline models.

	Viral				Top			
	1st Week	**2nd Week**	**3rd Week**	**4th week**	**1st Week**	**2nd Week**	**3rd Week**	**4th week**
Accuracy	0.9436	0.9375	0.9410	0.9353	0.9650	0.9824	0.9849	0.9827
AUC	0.9113	0.9073	0.9107	0.9161	0.9473	0.9784	0.9781	0.9806
MCC	0.8683	0.8496	0.8689	0.8558	0.9163	0.9568	0.9668	0.9613
Sensitivity	0.8269	0.8317	0.8214	0.8547	0.9029	0.9691	0.9561	0.9744
F1 Score	0.9005	0.8889	0.9020	0.9009	0.9394	0.9691	0.9776	0.9744
Precision	0.9885	0.9545	1.0000	0.9524	0.9789	0.9691	1.0000	0.9744
Specificity	0.9957	0.9830	1.0000	0.9776	0.9917	0.9877	1.0000	0.9869

Table 2: Performance of the models that do not use acoustic features.

	Viral				Top			
	1st Week	**2nd Week**	**3rd Week**	**4th week**	**1st Week**	**2nd Week**	**3rd Week**	**4th week**
Accuracy	0.8101	0.8185	0.7552	0.6706	0.8630	0.8798	0.8253	0.8121
AUC	0.7695	0.7714	0.6996	0.6087	0.7718	0.7886	0.7456	0.7264
MCC	0.5482	0.5582	0.4237	0.2333	0.6743	0.7030	0.6299	0.5816
Sensitivity	0.6635	0.6535	0.5357	0.4103	0.5437	0.5773	0.4912	0.4615
F1 Score	0.6832	0.6839	0.5911	0.4615	0.7044	0.7320	0.6588	0.6243
Precision	0.7041	0.7174	0.6593	0.5275	1.0000	1.0000	1.0000	0.9643
Specificity	0.8755	0.8894	0.8634	0.8072	1.0000	1.0000	1.0000	0.9913

Table 3: Performance of the models that use acoustic features only.

	Viral				Top			
	1st Week	**2nd Week**	**3rd Week**	**4th week**	**1st Week**	**2nd Week**	**3rd Week**	**4th week**
Accuracy	0.9080	0.8690	0.8584	0.7971	0.9417	0.9179	0.8554	0.8295
AUC	0.8510	0.7822	0.7857	0.7132	0.9057	0.8619	0.7895	0.7541
MCC	0.7871	0.6895	0.6868	0.5463	0.8614	0.7961	0.6888	0.6193
Sensitivity	0.7019	0.5644	0.5714	0.4444	0.8155	0.7320	0.5789	0.5214
F1 Score	0.8249	0.7215	0.7273	0.6012	0.8936	0.8353	0.7333	0.6740
Precision	1.0000	1.0000	1.0000	0.9286	0.9882	0.9726	1.0000	0.9531
Specificity	1.0000	1.0000	1.0000	0.9821	0.9958	0.9918	1.0000	0.9869

Table 4: Performance of the models that use all available data.

model quickly degrades for the 3rd and 4th weeks.

When we analyze all results, we verify that our proposed model has the highest metrics when only the list information is used to make the classification, with the exception of specificity and precision. Compare those metrics in Table 2 and Table 4 for the 1st and 2nd weeks. This result suggests that the past performance of a song is a good indicator of whether it will experience a spike in popularity, and that using acoustic features may improve the performance of the classifier, but is not required.

We note that the model based on the lists past information shows little degradation as the time window shifts away from the assessment phase. Notice in Table 2 that there is little reduction in all metrics for prediction in both lists. In the worst case, the precision dropped 3.6% in the "Virals" list and the accuracy dropped 0.8% in the same list. On the other hand, it is fair to say that, in spite of being consistent for all weeks during the prediction phase, it does not seem to substantially surpass the baseline in the first two weeks, as its accuracy and AUC are very similar to the baseline in the aforementioned rounds.

We do note that the model based exclusively in acoustic information obtained the worst results. Its performance was worse than the baseline according to many metrics for predictions in the "Most-Popular" list. We also note that this model degrades rapidly as the time window moves away from the assessment phase, declining 32.49% in MCC for predictions in "Virals". At best, the acoustic model dropped 7.8% in specificity for the "Virals" list and 0.8% in specificity for the "Most-Popular" list. Recall that specificity is the fraction of negative instances correctly identified. This suggests that the acoustic model tends to be over-optimistic in predicting that a song will go viral. Accuracy drops by approximately 12% in both lists and precision drops by approximately 7% and 3% in "Virals" and "Most-Popular", respectively.

The incorporation of acoustic information to the popularity model has proved to give less exciting results than expected. While the performance was not as low as when only acoustic information was used, it did not surpass the model that relies only on data from past "Virals" and "Most-Popular" lists. This is a surprising result, which deserves further analysis. We do note, however, that this model did not degrade as much as when only acoustic information was used, which do suggest that this model could be modified to properly predict success several days after the assessment phase has finished.

6 Conclusion and Future Work

In this paper we have discussed an approach to predict whether a song will have a spike in popularity (i.e., if it will "go viral") or if a song will be consistently popular. We define that a song has experienced a spike in popularity if it appears in Spotify's "Virals" list, and that a song is consistently popular if it is featured in Spotify's "Most-Popular" list. Our model predicts whether a song that is featured in one list will be featured in the other the next day. This approach has been chosen because it is generally more difficult for famous artists to suddenly experience a spike in popularity and be featured in the "Virals" list, while less successful artists tend to find it harder to appear in the "Most-Popular" list. We expect that to make our model useful in both situations.

There are several works in the literature that deal with the problem of predicting song or album success. Our approach has the advantage of requiring only data from the song past popularity. While acoustic information may be used, it is not necessary, and in fact our experiments have shown that results are superior when only previous popularity knowledge is used to train the model.

Because we did not find a suitable baseline for our experimentation model, we also propose a baseline in our paper. The proposed baseline considers how often a song has been featured in the "Most-Popular" and "Virals" list with respect to other songs, and predicts that they will be successful or experience a spike in popularity if they are more frequent than the median of the frequencies in the appropriate list. Our model was observed to be consistent when the prediction window is further from the training date, and achieved high statistics in all metrics chosen for evaluation.

We conclude that the proposed model is successful in employing popularity information to predict if a song will "go viral", and to predict if "viralization" phenomena will be followed by consistent growth in popularity for a given song.

We do note that, while our work is limited by to the extent of the data provided by Spotify, we believe our proposed model may be extended to other platforms. For example, some platforms provide lists of "trending" songs, or artists, which we could consider equivalent to Spotify's "Virals" list. On the other hand, streaming platforms usually provide lists of popular artists, albums, or songs, therefore they could be directly used by our model.

Furthermore, while we do aim at exploring models that do not require social network information, we note that our model might benefit from them. And we intend to use that type of data to garner more information as means of more accurately establishing the popularity of a song or artist.

This paper is part of a series related to music success prediction. On previous works we demonstrate the importance of social networks for the success of an album [8] and how the process of artist collaboration in Rap Music works [20]. We also identified influence factors on the popularity of musical genres [21].

References

[1] Chris Molanphy. How the hot 100 became america's hit barometer, 2013.

[2] Cristian Cibils, Zachary Meza, and Greg Ramel. Predicting a song's path through the billboard hot 100', 2015.

[3] Junghyuk Lee and Jong-Seok Lee. Predicting music popularity patterns based on musical complexity and early stage popularity. In *Proceedings of the Third Edition Workshop on Speech, Language &#38; Audio in Multimedia*, SLAM '15, pages 3–6, New York, NY, USA, 2015. ACM.

[4] Ioannis Karydis, Aggelos Gkiokas, Vassilis Katsouros, and Lazaros Iliadis. Musical track popularity mining dataset: Extension & experimentation. *Neurocomputing*, 280:76 – 85, 2018. Applications of Neural Modeling in the new era for data and IT.

[5] Vasant Dhar and Elaine A. Chang. Does chatter matter? the impact of user-generated content on music sales. *Journal of Interactive Marketing*, 23(4):300 – 307, 2009.

[6] Benjamin Shulman, Amit Sharma, and Dan Cosley. Predictability of popularity: Gaps between prediction and understanding. In *Tenth International AAAI Conference on Web and Social Media*, pages 348–357, 2016.

[7] Yekyung Kim, Bongwon Suh, and Kyogu Lee. #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*, SoMeRA '14, pages 51–56, New York, NY, USA, 2014. ACM.

[8] Carlos V.S. Araujo, Rayol M. Neto, Fabiola G. Nakamura, and Eduardo F. Nakamura. Predicting music success based on users' comments on online social networks. In *Proceedings of the 23rd Brazillian Symposium on Multimedia and*

the Web, WebMedia '17, pages 149–156, New York, NY, USA, 2017. ACM.

[9] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.

[10] Shushan Arakelyan, Fred Morstatter, Margaret Martin, Emilio Ferrara, and Aram Galstyan. Mining and forecasting career trajectories of music artists. In *Proceedings of the 29th on Hypertext and Social Media*, HT '18, pages 11–19, New York, NY, USA, 2018. ACM.

[11] Dennis M Steininger and Simon Gatzemeier. Using the wisdom of the crowd to predict popular music chart success. In *Proceedings of the 21st European Conference on Information Systems*, page 215, 2013.

[12] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, pages 18–25, 2015.

[13] S. Dubnov. Generalization of spectral flatness measure for non-gaussian linear processes. *IEEE Signal Processing Letters*, 11(8):698–701, Aug 2004.

[14] David Harris and Sarah Harris. *Digital design and computer architecture*. Morgan Kaufmann, 2010.

[15] Jason Brownlee. *Introduction to Time Series Forecasting with Python: How to Prepare Data and Develop Models to Predict the Future*. Jason Brownlee, 2017.

[16] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[17] R. Giusti, D. F. Silva, and G. E. A. P. A. Batista. Improved time series classification with representation diversity and svm. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1–6, Dec 2016.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[19] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[20] Carlos V.S. Araújo, Rayol M. Neto, Fabiola G. Nakamura, and Eduardo F. Nakamura. Using complex networks to assess collaboration in rap music: A study case of dj khaled. In *Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web*, WebMedia '17, pages 425–428, New York, NY, USA, 2017. ACM.

[21] Carlos V. S. Araujo and Eduardo F. Nakamura. Identification of most popular musical genres and their influence factors. In *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web*, WebMedia '18, pages 233–236, New York, NY, USA, 2018. ACM.