A Model for Predicting Music Popularity on Spotify

Carlos Vicente Soares Araujo
Loggi
vicente@icomp.ufam.edu.br

ABSTRACT

The global music market moves billions of dollars every year, most of which comes from streaming platforms. In this paper, I present a model for predicting whether or not a song will appear in Spotify's Top 50 ranking. To make this prediction, I trained different classifiers with information of 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 and AUC above 80%.

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 was even higher accounting for 56.1% of global music revenues [2]. Therefore, streaming has become critical for artists 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. 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.

In this paper, I present a model to predict if a song will be popular on Spotify streaming platform even before its release. Spotify was chosen as my study case because it is one of the world's largest music streaming service in number of users [3].

2. METHODOLOGY

For the remainder of this paper, I will adopt the acronym PM when dealing with my Proposed Model.

The data collection was performed using the Spotify Web API [4]. From November 2018 to July 2019 I 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 [5]. In this work, I consider the songs in the Top 50 as popular and the ones on Viral 50 as unpopular, an approach also used on similar works [1].

For each song, I collected the names of the artists, an Explicit flag, the songs ID's, and audio features as: danceability, energy, speechiness, acousticness, instrumentalness, liveness and valence. All of these audio features are float fields and the documentation does not tell how they are calculated. Therefore, I 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, I 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 I used the values 0.33 and 0.8 as a basis, respectively, due to the description of these fields in the documentation.

For my experiments, I 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, I 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, I established that for a song to be considered popular it should have appeared more than four times in the Top 50.

For comparison, I set up a model based on the methodology used by Reiman and Örnell [2]. I will use the acronym ROM (Reiman and Örnell Model) when dealing with this model from now on. To make it, I had to collect others audio features from Spotify's API, they are: dura-

[1] https://www.ifpi.org/downloads/GMR2019.pdf
[2] https://bit.ly/2RKR56L
[3] http://bit.ly/2KwJmGu
[4] https://spoti.fi/37vPA2l
[5] 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.

Table 1. 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 2. 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

tion, key, mode, tempo and time signature. ROM do not uses the Explicit flag.

For ROM, I did not performed the binarization process and I did not normalized the data neither. I also set up two databases for ROM in order to compare against the results obtained with my methodology. The instances of these two databases represent the same entries as the PM ones.

I used different machine learning algorithms in my experiments. Therefore, it was necessary to divide the databases into training and testing groups. In the first experiment, I 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.

To make the results more comparable, I restricted the number of algorithms used in my experiments to those that were also used by Reiman and Örnell [2]. Thus, the algorithms used were Gaussian Naive Bayes (GNB), K-Nearest Neighbors (KNN), Logistic Regression (LR) and Support Vector Machine (SVM) with RBF kernel.

To evaluate the results obtained I used the following metrics: accuracy, precision, Negative Predictive Value (NPV), recall, specificity, F1 Score, Area Under the Receiver Operating Characteristic Curve (AUC) and Matthews Correlation Coefficient (MCC).

3. RESULTS

Table 1 shows the values achieved in the evaluation metrics in the experiment where the predictions were made on a per-day basis. While, Table 2 presents the values of the experiment where the predictions were made on a per-song basis. The best results obtained in each of the metrics are shown in red.

In the two experiments, MP reached the best results when SVM classifier was used, while KNN was the best for ROM. In comparison, PM got better results in all metrics in the two experiments as seen in Table 3.

Table 3. 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%

4. FUTURE WORK

As future work my idea is to also use data from social networks, as a previous work of mine 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 on Twitter [3].

5. REFERENCES

[1] D. Herremans, D. Martens, and K. Sörensen, "Dance hit song prediction," *Journal of New Music Research*, vol. 43, no. 3, pp. 291–302, 2014. [Online]. Available: https://doi.org/10.1080/09298215.2014.881888

[2] M. Reiman and P. Örnell, "Predicting hit songs with machine learning," ser. TRITA-EECS-EX, no. 2018:202, 2018.

[3] C. V. Araujo, R. M. Neto, F. G. Nakamura, and E. 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*, ser. WebMedia '17. New York, NY, USA: ACM, 2017, pp. 149–156. [Online]. Available: http://doi.acm.org/10.1145/3126858.3126885