

# Will I Remain Popular? A Study Case on Spotify

Carlos V. S. Araujo<sup>1</sup>, Marco A. P. Cristo<sup>1</sup>, Rafael Giusti<sup>1</sup>

<sup>1</sup>Instituto de Computação  
Universidade Federal do Amazonas (UFAM) – Manaus, AM – Brasil

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

**Abstract.** *Online streaming platforms are now the most important form of music consumption. In this paper, we present a model for predicting if a popular song on Spotify will remain popular after a certain amount of time. Spotify is the second biggest global streaming service. If a song is popular on this platform it will ensure a good financial return for the artist and his label. We approach the problem as a classification task and employ classifiers built on past information from the platform's Top 50 Global ranking. The Support Vector Machine with linear kernel classifier reached the best results. We also verify if acoustic information can provide useful features for this problem. We made a series of classification rounds, where the results of one round were used as input of posterior rounds. Our results show that rankings previous data alone is sufficient to predict if a song will remain at the Top 50 Global two months in advance, achieving accuracy, negative predictive value, recall, specificity and F1 Score higher than 70% for this task. Moreover, results up to 6.60% higher were obtained using songs acoustic information.*

**Resumo.** *Atualmente, as plataformas de streaming on-line são a forma mais importante de consumo de música. Neste artigo, apresentamos um modelo para prever se uma música popular no Spotify permanecerá popular após um determinado período de tempo. O Spotify é o segundo maior serviço global de streaming. Se uma música for popular nessa plataforma, ela garantirá um bom retorno financeiro para o artista e sua gravadora. Abordamos o problema como uma tarefa de classificação e empregamos classificadores baseados em informações prévias do ranking Top 50 Global da plataforma. O classificador de Máquina de Vetores de Suporte (SVM) com kernel linear alcançou os melhores resultados. Também verificamos se informações acústicas podem fornecer recursos úteis para esse problema. Fizemos uma série de rodadas de classificação, onde os resultados de uma rodada eram utilizados como entrada da rodada posterior. Nossos resultados mostram que os dados prévios são suficientes para prever se uma música permanecerá no Top 50 Global com dois meses de antecedência, alcançando precisão, valor preditivo negativo, revocação, especificidade e valor-F1 acima de 70% para essa tarefa. Ademais, resultados até 6,60% superiores puderam ser obtidos ao utilizar as características acústicas das faixas.*

## 1. Introdução

O mercado musical movimentou bilhões de dólares todos os anos. Em 2018 o faturamento desse mercado foi de US\$ 19.1 bilhões segundo a International Federation of the

Phonographic Industry (IFPI) – em português, Federação Internacional da Indústria Fonográfica. Esse montante considera somente ganhos provenientes de vendas físicas e digitais, *streaming*, direitos de apresentações e receitas de sincronização (como as provenientes de rádios e do uso em filmes, séries e afins). Desse total, 47% são advindos de consumo via *streaming* e 12% de vendas digitais<sup>1</sup>.

Os dois maiores provedores de *streaming* de música em número de usuários – NetEase e Tencent Holdings’ – são exclusivos do mercado chinês. Daqueles que oferecem serviços globalmente, os dois maiores são o Soundcloud e o Spotify, sendo que o primeiro é voltado para artistas independentes e/ou iniciantes<sup>2,3</sup>. Logo, obter grande quantidade de *streams* no Spotify pode garantir bons resultados comerciais. Uma forma de verificar quais músicas se destacam na plataforma é através do ranking Top 50 Global, que conta com as 50 músicas mais ouvidas no Spotify no dia anterior ao da publicação da lista.

Neste trabalho apresentamos um modelo baseado em aprendizagem de máquina para prever se uma música que já apareceu no Top 50 Global do Spotify irá nele permanecer após um determinado período de dias. Esse modelo é importante pois pode ajudar artistas e gravadoras a identificar se uma música trará bons resultados comerciais em longo prazo, visto que esse ranking é um indicativo do mercado. Assim, caso o modelo mostre que o resultado esperado não será obtido, então artistas e gravadoras podem tomar estratégias de marketing para contornar tal situação.

O problema de realizar predições no mercado musical vem sendo estudado na literatura, onde algoritmos de aprendizagem de máquina apresentam melhores resultados de forma geral. Dados de diferentes fontes foram utilizados para desenvolver modelos preditivos, como informações de redes sociais [Dhar and Chang 2009] [Shulman et al. 2016], das características acústicas das faixas [Lee and Lee 2018] [Interiano et al. 2018], de shows e festivais [Arakelyan et al. 2018] [Steininger and Gatzemeier 2013] e de colaborações entre artistas [Silva et al. 2019] [Araújo et al. 2017]. Para o desenvolvimento de nosso modelo utilizamos duas abordagens com o intuito de identificar qual destas apresenta melhor resultado em nossos experimentos. A primeira utiliza somente dados históricos do Top 50 Global, enquanto a outra usa também de informações das características acústicas das faixas.

O restante deste trabalho está organizado da seguinte forma: na Seção 2 mostramos trabalhos relacionados ao realizado nesse artigo. Enquanto, na Seção 3 a metodologia que utilizamos é apresentada. Os resultados obtidos estão na Seção 4. Por fim, na Seção 5 fazemos uma breve conclusão e apontamos possíveis trabalhos futuros.

## 2. Trabalhos Relacionados

Diferentes abordagens já foram utilizadas em estudos de previsão do sucesso no mercado musical. Dentre elas destacamos quatro abordagens. A primeira consiste em utilizar dados de redes sociais para estimar a popularidade de músicas e artistas. A segunda consiste em extrair características das próprias músicas como principal fonte de informação. A ter-

---

<sup>1</sup>[www.ifpi.org/downloads/GMR2019.pdf](http://www.ifpi.org/downloads/GMR2019.pdf)

<sup>2</sup><https://www.forbes.com/sites/hughmcintyre/2018/05/25/the-top-10-streaming-music-services-by-number-of-users/#707898eb5178>

<sup>3</sup><https://www.fool.com/investing/2018/12/17/whats-most-popular-music-streaming-service-2018.aspx>

ceira foca no uso de informações sobre concertos e festivais para realizar suas previsões. Enquanto, a quarta baseia-se no uso de dados prévios de *rankings* e listas de popularidade para o desenvolvimento de modelos de previsão. Nessa seção apresentaremos quatro trabalhos, onde essas abordagens foram aplicadas.

[Abel et al. 2010] coletaram, através da plataforma Spinn3r<sup>4</sup>, postagens em *blogs* sobre 50 álbuns entre os dias 20 de maio a 31 de julho de 2009. Os autores também coletaram informações sobre a posição dos álbuns nos *rankings* de álbuns mais vendidos na Amazon nesse mesmo período. Os autores buscavam prever se determinado álbum iria cair, subir ou ficar na mesma posição no *ranking* com base no volume de postagens nos dias anteriores ao da predição, no gênero da música e no número de dias transcorridos do lançamento do álbum até a data de predição. Os autores obtiveram acurácia de 49,70% no melhor caso, utilizando o algoritmo de Tabela de Decisão para realizar a previsão.

Já [Interiano et al. 2018] utilizaram características acústicas das músicas para realizar suas predições. Os autores coletaram dados de mais de 500.000 músicas a partir do Top 100 de *Singles* no Reino Unido da Official Charts Company (OCC)<sup>5</sup> e do site MusicBrainz<sup>6</sup>. A presença de uma música no Top 100 foi considerada como indicador de sucesso de uma música nessa pesquisa. Utilizando o AcousticBrainz<sup>7</sup>, os autores extraíram as características acústicas das faixas da base, incluindo variáveis binárias que indicam se estas são alegres, tristes, dançantes e/ou agressivas. Utilizando-se dessas informações, o objetivo era de determinar se uma música qualquer seria ou não um sucesso. Utilizando o classificador Random Forest obtiveram acurácia de 70%. Ademais, os autores também realizaram experimentos nos quais adicionavam um campo à base de dados indicando se a música teve a contribuição de um *superstar* – definido pelos autores como um artista ou banda que teve ao menos uma música na primeira posição do *ranking* nos cinco anos anteriores ao da pesquisa. O acréscimo desse campo permitiu a obtenção de um modelo com acurácia de 85%.

Em relação ao uso de dados de eventos para realização da previsão no mercado, destacamos o trabalho de [Arakelyan et al. 2018]. Utilizando dados do site SongKick<sup>8</sup>, os autores coletaram informações sobre 645.507 concertos, incluindo a localização e nome do evento, a lista de artistas participantes e um valor em popularidade dado pela plataforma. Nesse estudo, um artista era considerado popular caso possuísse contrato com uma das seguintes gravadoras ou seus selos afiliados: Sony BMG, Universal Music Group, Warner. Os autores buscavam prever a partir das informações dos concertos quais artistas tinham contrato com essas gravadoras. Para realizar essa previsão foi utilizado o método de regressão logística. A precisão máxima obtida foi de 39%, entretanto o valor em AUC foi de 87%.

A abordagem que mais se assemelha à desta pesquisa é empregada por [Herremans et al. 2014]. Os autores utilizaram somente dados do Top 40 de músicas do gênero *dance* da OCC em seu estudo. Os dados foram coletados do período de outubro de 2009 a março de 2013, totalizando um total de 7.159 entradas individuais, e 759

---

<sup>4</sup><http://docs.spinn3r.com/>

<sup>5</sup><https://www.officialcharts.com/>

<sup>6</sup><https://musicbrainz.org/>

<sup>7</sup><https://acousticbrainz.org/>

<sup>8</sup><https://www.songkick.com/>

músicas distintas. Esse valor é menor que o total de entradas possíveis para o período, pois somente foram consideradas aquelas em que foram possíveis extrair as seguintes informações: nome da música, artista, posição e data. Os autores consideravam uma música como sucesso se ela estivesse até uma determinada posição da lista. Os melhores resultados foram alcançados quando o critério para o sucesso foi a música estar no Top 10, enquanto aquelas entre as posições 31 e 40 foram dadas como fracassos e as músicas nas posições 11 a 30 foram descartadas desse experimento. Os autores conseguiram obter 85% de acurácia média utilizando o classificador de Máquina de Vetores de Suporte (SVM) com *kernel* polinomial, sendo a base dividida em diferentes conjuntos de treino e teste utilizando validação cruzada.

Nosso trabalho se insere na categoria daqueles que se utilizam, principalmente, de dados de *rankings* para realizar as previsões. A principal distinção do nosso trabalho para os anteriores é que realizamos várias etapas de treinamento e avaliação para obter previsões progressivamente mais distantes do período no qual os dados foram coletados. Dessa maneira, conseguimos prever com dois meses de antecedência se uma música será ou não popular. Em nossos experimentos, também fazemos uso das características acústicas, visando principalmente comparar a eficácia dos modelos preditivos quando utilizamos e não utilizamos tais características.

### 3. Metodologia

Uma vez que o nosso objetivo é prever a popularidade futura de uma música, um dos pontos-chave deste estudo é estabelecer um conceito de popularidade. Optamos por estabelecer que uma música *está* popular se ela aparece na lista “Top 50” global da plataforma de *streaming* Spotify. Essa abordagem é semelhante às de trabalhos relacionados que definem que uma obra é popular se aparece em algum *ranking* ou lista conceituada. Entretanto, isso nos traz uma limitação da própria plataforma: a WebAPI que utilizamos para coleta (Subseção 3.1) só fornece dados para músicas *enquanto* elas estão presentes no *ranking*. Um possível contorno para esse problema seria construir um conjunto de músicas não populares com base em *ranking* anteriores.

Como mencionado anteriormente, observamos que estratégias baseadas em aprendizagem de máquina (AM) tendem a produzir melhores resultados, de maneira geral. Assim, nossa metodologia será descrita em torno das etapas típicas de um processo de AM: coleta e pré-processamento de dados (Subseção 3.1), construção do modelo (Subseção 3.2) e avaliação dos resultados (Subseção 3.3). A principal diferença do nosso trabalho com respeito aos que verificamos durante nosso mapeamento da literatura é que realizamos diversas etapas de indução e classificação para expandir nosso horizonte de previsão. Uma representação gráfica de nossa metodologia, que será detalhada nas próximas subseções, pode ser encontrada na Figura 1.

#### 3.1. Coleta e pré-processamento dos dados

Por meio da WebAPI do Spotify<sup>9</sup>, a cada dia podemos coletar uma edição diferente do *ranking* “Top 50” dessa plataforma. Cada edição desse *ranking* contém 50 entradas correspondentes às músicas mais ouvidas no dia anterior.

---

<sup>9</sup><https://developer.spotify.com/documentation/web-api/>

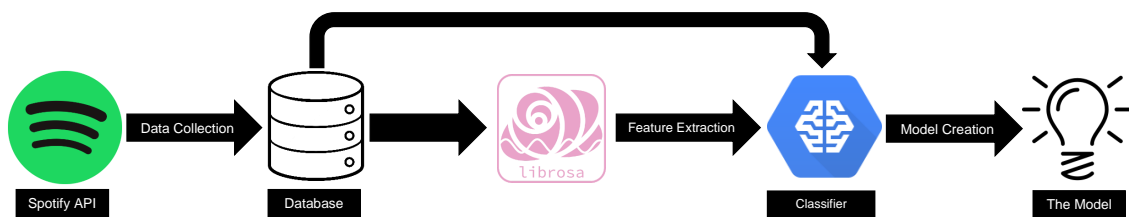


Figura 1. Representação da metodologia utilizada.

Cada instância do nosso conjunto de dados será extraída de uma entrada de uma edição do *ranking*. Ou seja, uma música em particular pode estar associada a várias instâncias. Como buscamos realizar previsões para um período de tempo de longo prazo em relação aos dados de entrada, é importante que o classificador aprenda o comportamento dessas informações ao longo do tempo. Para isso, cada instância é aumentada com variáveis binárias que indicam a presença dessas músicas em dias posteriores do *ranking*. Tais campos são conhecidos como campos *lagged* [Brownlee 2017] e permitem que um problema de natureza inerentemente temporal possa ser abordado através de um modelo não necessariamente temporal. Optamos por utilizar essa abordagem, pois apenas temos informações sobre as músicas *enquanto* elas estão no *ranking*, de modo que coletar as instâncias como séries temporais produziria séries curtas e com muitos valores faltantes.

Como o nosso objetivo é prever se uma música fará sucesso – o que definimos como estar presente no *ranking* “Top 50” do Spotify –, o rótulo de cada instância é uma variável binária que indica se a música estará presente no *ranking* um certo número de dias no futuro. Essa isonomia entre a classe e as variáveis *lagged* nos permitem realizar várias rodadas de classificação a fim de expandir o horizonte de previsão. Em cada rodada, os campos *lagged* são utilizados para estimar dados reais na rodada seguinte. Na Subseção 3.2 explicamos como realizamos tal processo.

Para cada entrada em uma edição do *ranking*, coletamos informações de nove campos diretamente da API. São eles: a posição da música no *ranking* (*i.e.*, o *rank*), a data do *ranking*, os nomes dos artistas e da faixa, a data de lançamento da canção e sua duração em milissegundos. Há também um indicador de popularidade, definido pela plataforma como um valor inteiro no intervalo  $[0, 100]$ , e um campo binário que indica se a música contém palavras de baixo calão. Além disso, cada entrada possui uma URL para uma amostra de 30 segundos da música, que utilizamos para extrair características acústicas—a URL em si não é armazenada como um atributo.

Realizamos a coleta no período de 01/11/2018 a 29/05/2019. Notamos que cerca de 3% das músicas observadas nesse período não dispunham de uma URL. Devido ao baixo número e visando homogeneidade em todas as abordagens, decidimos remover todas as instâncias que não contivessem uma URL válida. Utilizando o pacote para Python LibROSA [McFee et al. 2015] extraímos cinco características acústicas, descritas a seguir:

1. **Mel-Frequency Cepstral Coefficients (MFCC):** obtido a partir do espectro da representação mel comprimida do sinal. O MFCC é provavelmente um dos recursos mais usados no processamento de fala e é uma representação expressiva de baixa dimensão de um sinal. Neste trabalho usamos 13 coeficientes por música;
2. **Centroide Espectral:** o centroide de cada quadro de um espectrograma de mag-

nitude que foi normalizado e tratado como uma distribuição sobre intervalos de frequência;

3. **Coefficiente de Tonalidade:** uma medida de quão parecido a um ruído um som é, isto é, como oposto a ser próximo a um tom [Dubnov 2004];
4. **Passagem por Zero:** o número de vezes que uma forma de onda muda de sinal;
5. **Tempo:** número de batidas por minuto.

Por fim, adicionamos a cada instância a data da primeira ocorrência da música no *ranking* e realizamos o pré-processamento dos dados. Usando a biblioteca de manipulação de dados Pandas [McKinney 2010] os nomes das músicas e dos artistas foram transformados em categorias utilizando *one-hot encoding* [Harris and Harris 2010]. Além disso, todos os campos que representam datas foram transformados em valores inteiros utilizando a função “`to_numeric`”<sup>10</sup>.

Por fim, realizamos um processo de padronização do conjunto de dados, na qual os valores numéricos são substituídos por seus escores padrões (*z-scores*). Isso é feito removendo a média e dimensionando os dados para a variância de cada coluna. Essa etapa é importante pois estimadores de aprendizagem de máquina comportam-se mal se as características individuais não se pareçam com um padrão de distribuição normal.

### 3.2. Predição dos Dados

Após a coleta e o pré-processamento dos dados, podemos realizar o treinamento de um modelo. É importante salientar que o modelo é muito influenciado pelos atributos *lagged* que acrescentamos. O número  $k$  de campos e o tamanho  $X$  da janela, em dias, definem o período no qual uma instância representa uma música. Por exemplo, se empregarmos  $k=3$  atributos com janela de tamanho  $X=7$  dias, então uma instância conterá o *rank* da música na data em que foi observada e, além disso, a popularidade dessa música 7, 14 e 21 dias depois da data do *ranking*. A classe pode ser referente à popularidade da música em qualquer data *após* esses 21 dias, mas nós empregamos o mesmo fator  $X$  para estabelecer essa data, de modo que a classe será a popularidade da música 28 dias depois de ter sido observada no *ranking*.

Nós treinamos os modelos com 40 edições do *ranking*. Buscamos neste trabalho encontrar a distribuição de janelas e a quantidade de campos *lagged* que maximizasse a janela em que seria feita a predição, mas cujos resultados pudessem ser comparados aos dados reais. Isso porque utilizar mais variáveis e janelas maiores permite definir a classe com maior distância em relação a data de coleta da instância, o que é desejável, mas demanda mais tempo de coleta para compor os dados de treinamento (o Spotify não permite acesso a edições passadas do *ranking*). Assim, considerando os dados que havíamos coletado, estabelecemos  $k=3$  e  $X=20$ .

As instâncias do conjunto inicial de treinamento foram extraídas de 40 edições do *ranking* entre 1º de novembro de 2018 e 11 de dezembro do mesmo ano. Cada instância recebeu variáveis *lagged* binárias correspondentes à popularidade da música 20, 40 e 60 dias após a edição do *ranking*, enquanto a classe indicava a popularidade da música 80 dias depois da edição do *ranking*. Por exemplo, uma instância extraída do *ranking* de

---

<sup>10</sup>[https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.to\\_numeric.html](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.to_numeric.html)

1º de novembro recebeu os campos *lagged* de 21 de novembro, 11 de dezembro e 31 de dezembro e a classe indicava a presença dessa música no *ranking* do dia 20 de janeiro de 2019.

Em síntese, cada instância extraída a partir de uma edição específica do *ranking* possui como classe o campo binário que indica a sua popularidade 80 dias depois, mas apenas 20 dias após o último atributo *lagged*. Para contornar essa limitação e expandir o horizonte de previsão, nós realizamos várias rodadas preditivas, nas quais estimamos atributos *lagged* para datas cada vez mais distantes da data de coleta.

Uma síntese das rodadas que realizamos pode ser visualizada na Figura 2. Nela podem ser vistas as datas das quais as informações foram extraídas. Foram realizadas três rodadas de classificação dos dados. Em cada uma dessas rodadas nós treinamos um classificador para prever 20 dias além dos dados de entrada. As variáveis alvo (classes) previstas em cada rodada eram utilizadas como entradas nas rodadas seguintes. Logo, a partir da segunda rodada eram utilizados dados sintéticos junto aos reais como entradas no classificador. Na última rodada, o atributo-alvo é a classe desejada, indicando se uma música será popular ou não dois meses após o término da coleta de dados.

	Dados Reais						Dados Sintéticos		
	Dia 01	21	41	61	81	101	121	141	161
1ª Rodada	Real	Real / Lagged	Lagged		Lagged / Alvo	Alvo			
			Real	Lagged			Alvo		
2ª Rodada		Real	Real / Lagged	Lagged		Lagged / Alvo	Alvo		
				Real	Lagged			Alvo	
3ª Rodada			Real	Real / Lagged	Lagged		Lagged / Alvo	Alvo	
					Real	Lagged			<b>Classe</b>

**Figura 2. Descrição das rodadas de classificação.**

Uma quarta rodada seria possível de ser realizada, na qual os valores previstos entre os dias 161 e 180 seriam utilizados como atributos *lagged* para estender o horizonte preditivo em mais vinte dias. Entretanto, nosso processo de coleta de dados foi interrompido nos primeiros dias de maio de 2019 e, portanto, o experimento foi finalizado em três rodadas. Note, entretanto, que a metodologia proposta neste trabalho funcionaria, em princípio, para qualquer número arbitrário de rodadas, dependendo do número de variáveis *lagged* estabelecidas e do tamanho da janela.

### 3.3. Avaliação Experimental

Embora no processo de classificação repetida tenhamos gerado vários modelos, nosso interesse está em verificar a capacidade de realizar previsões em longo prazo. Portanto, o conjunto de teste foi construído com base nos *rankings* utilizados na etapa de classificação da última rodada. Ou seja, utilizando como atributos os dados reais dos dias 81-100 e os atributos *lagged* inferidos nas rodadas anteriores, queríamos testar a eficácia de modelos de aprendizagem de máquina para prever a popularidade das músicas de acordo com os *rankings* dos dias 161-180.

Nós utilizamos dois conjuntos de dados para treinar nossos modelos. Um deles contendo somente os dados prévios das listas, enquanto o outro abrangendo também as informações das características acústicas das faixas.

Os modelos foram treinados e testados utilizando a biblioteca “scikit-learn” [Pedregosa et al. 2011]. Nessa etapa utilizamos de um pipeline, onde um primeiro modelo foi utilizado para selecionar os campos de maior importância na base. Nessa etapa utilizamos um SVM com *kernel linear*. Após isso, os dados passavam para um segundo modelo que era responsável pela classificação em si. Nessa segunda etapa foram utilizados oito diferentes classificadores, são eles: Ada Boost, Bernoulli e Gaussian Naive Bayes (NB), Random Forest (RF) e SVM com *kernel linear*, polinomial, RBF e sigmoid. Mantivemos os valores padrão recomendados para todos os parâmetros.

Para avaliação dos resultados alcançados, a partir das matrizes de confusão geradas pelos modelos computamos nove diferentes métricas de avaliação. Tais métricas são apresentadas na sequência, com explanação baseada em [Olson and Delen 2008]:

1. **Acurácia:** a porcentagem de instâncias corretamente previstas;
2. **Precisão:** a porcentagem de valores positivos corretamente previstos;
3. **Valor Preditivo Negativo (VPN):** a porcentagem de valores negativos corretamente previstos;
4. **Revocação:** a porcentagem de positivos reais;
5. **Especificidade:** a porcentagem de negativos reais;
6. **Valor-F1:** a média harmônica entre a precisão e revocação;
7. **Área Abaixo da Curva ROC (AUC):** probabilidade do classificador classificar uma instância positiva escolhida aleatoriamente mais alta do que uma negativa aleatoriamente escolhida, quando utiliza-se de unidades normalizadas [Fawcett 2006];
8. **Fall-out:** a porcentagem de falsos positivos, quanto menor melhor é o resultado;
9. **Taxa de Perda:** a porcentagem de falsos negativos, quanto menor melhor é o resultado.

#### 4. Resultados

Em nossa base, 609 instâncias entre os dias 20 de janeiro a 08 de fevereiro não apareceram nas listas de 10 a 29 de abril, enquanto outras 350 permaneceram no ranking. Relembrando que não há um total de 1000 instâncias, pois algumas destas foram removidas da base por não conter a URL para o trecho de 30 segundos da música.

Na Tabela 1 estão os valores calculados das métricas de avaliação para os modelos que utilizam somente os dados prévios das listas. Enquanto, para aqueles que usam também das informações de características acústicas das faixas é possível ver as suas performances na Tabela 2.

O classificador Gaussian Naive Bayes foi o pior em nosso experimento. Mas, essa situação era esperada, visto que o foco dos algoritmos Naive Bayes está na performance, isto é, na rápida execução de sua classificação [Zhang 2004]. Entretanto, os resultados obtidos por esses classificadores são importantes, pois podem servir como *baseline*.

De forma geral, o melhor modelo em nossa pesquisa foi treinado utilizando todos os dados em um classificador SVM com *kernel linear*. Seus resultados foram 97,35% superiores em acurácia e 260,29% em especificidade em relação aos obtidos pelo Gaussian



**Tabela 1. Performance dos modelos que utilizam somente dos dados prévios do Top 50**

Dados Prévios	Ada Boost	Bernoulli NB	Gaussian NB	Random Forest	SVM Linear	SVM Poli	SVM RBF	SVM Sigmoid
Acurácia	0.6924	0.7278	0.5172	0.6903	0.7372	0.6455	0.7143	0.7049
Precisão	0.5641	0.5945	0.3975	0.5978	0.5984	0.5146	0.5757	0.5708
VPN	0.7962	0.8566	0.6789	0.7267	0.8872	0.7180	0.8665	0.8354
Revocação	0.6914	0.8000	0.6257	0.4629	0.8514	0.5029	0.8257	0.7714
Especificidade	0.6929	0.6864	0.4548	0.8210	0.6716	0.7274	0.6502	0.6667
Valor-F1	0.6213	0.6821	0.4861	0.5217	0.7028	0.5087	0.6784	0.6561
AUC	0.6922	0.7432	0.5403	0.6419	0.7615	0.6151	0.7380	0.7190
Fall-out	0.3071	0.3136	0.5452	0.1790	0.3284	0.2726	0.3498	0.3333
Taxa de Perda	0.3086	0.2000	0.3743	0.5371	0.1486	0.4971	0.1743	0.2286

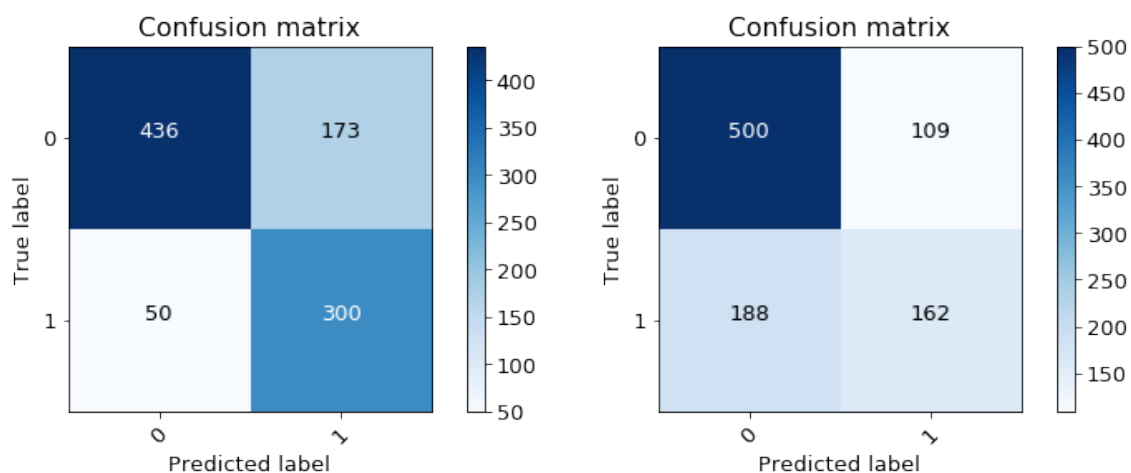
**Tabela 2. Performance dos modelos que utilizam todos os dados disponíveis**

Todos os Dados	Ada Boost	Bernoulli NB	Gaussian NB	Random Forest	SVM Linear	SVM Poli	SVM RBF	SVM Sigmoid
Acurácia	0.7372	0.7247	0.3889	0.6538	0.7675	0.6945	0.7143	0.6840
Precisão	0.6099	0.5915	0.3405	0.5313	0.6342	0.5899	0.5805	0.5512
VPN	0.8480	0.8528	0.5525	0.7064	0.8971	0.7461	0.8439	0.8060
Revocação	0.7771	0.7943	0.7200	0.4371	0.8571	0.5343	0.7829	0.7229
Especificidade	0.7143	0.6847	0.1987	0.7783	0.7159	0.7865	0.6749	0.6617
Valor-F1	0.6834	0.6780	0.4624	0.4796	0.7290	0.5607	0.6667	0.6255
AUC	0.7457	0.7395	0.4593	0.6077	0.7865	0.6604	0.7289	0.6923
Fall-out	0.2857	0.3153	0.8013	0.2217	0.2841	0.2135	0.3251	0.3383
Taxa de Perda	0.2229	0.2057	0.2800	0.5629	0.1429	0.4657	0.2171	0.2771

Naive Bayes. A superioridade mínima encontrada entre esses dois classificadores foi de 19,04% em revocação. A matriz de confusão obtida pelo melhor modelo está apresentada na Figura 3a.

Observa-se por essa matriz que o maior ponto a se desenvolver nesse modelo é o dos falsos positivos, isto é, instâncias que o modelo prevê que estariam presentes na lista, mas que na realidade não estão. Tal situação explica o fato desse modelo não apresentar o melhor valor em especificidade em nosso experimento. Ele foi superado pelos classificadores RF e SVM com *kernel* polinomial treinados tanto com todos os dados, quanto somente com os dados prévios. O melhor resultado obtido segundo essa métrica foi utilizando somente os dados prévios treinados no classificador RF, a matriz de confusão obtida nesse caso está na Figura 3b. Observamos por essa matriz que esse classificador previu 71,74% das instâncias como negativas, o que diminuiu o número de falsos positivos, aumentando o valor em especificidade.

O alto valor em especificidade obtido pelo RF utilizando somente os dados prévios poderia ser um indicativo para sua escolha como melhor modelo. Entretanto, devido as particularidades de nosso caso de teste onde há 74% instâncias negativas a mais que positivas, então as métricas revocação e AUC são as mais relevantes. Comparativamente,



(a) Matriz de confusão para o modelo treinado com todos os dados utilizando classificador SVM com *kernel* linear.

(b) Matriz de confusão para o modelo treinado somente com dados prévios utilizando o classificador Random Forest.

Figura 3. Matrizes de confusão (a) do melhor modelo segundo sete das métricas utilizadas e (b) do melhor modelo segundo os valores de especificidade e *fall-out*.

o modelo SVM com *kernel* linear treinado com todos os dados obteve revocação e AUC 85,16% e 22,53% superiores em relação ao RF, respectivamente.

Ao utilizar as características acústicas das faixas, o modelo SVM com *kernel* linear apresentou uma pequena melhoria de 5,98% e 6,60% em precisão e especificidade, respectivamente. Uma possível explicação para esse aumento reduzido pode ser pelo fato dessas informações terem sido extraídas a partir de amostras de 30 segundos das músicas. Tais partes podem não ser representativas das canções em sua totalidade, o que pode ter enviesado tais dados.

Em síntese, conseguimos prever se uma música presente no Top 50 Global do Spotify irá ou não aparecer nesse mesmo ranking após dois meses com acurácia de 76,75%. Além disso, os valores obtidos em VPN, revocação, especificidade, Valor-F1 e AUC ficaram todos acima de 70%. Para obter tais resultados usamos um classificador SVM com *kernel* linear utilizando dados prévios do Top 50 e informações sobre as características acústicas das faixas da base.

## 5. Conclusão e Trabalhos Futuros

Nesse artigo apresentamos um modelo para prever se uma música presente no ranking Top 50 Global do Spotify irá nele permanecer após uma determinada janela de tempo. Tal modelo pode ajudar artistas a preverem se suas músicas trarão retorno financeiro a longo prazo, visto que são dos serviços de *streaming* que 47% dos ganhos do mercado musical são advindos. A escolha do Spotify foi dada pois esse é o maior serviço de *streaming* com disponibilidade global e que contém músicas de artistas e gravadoras de renome.

Há uma gama de trabalhos na literatura que tratam do problema de prever o sucesso de músicas e álbuns. Nesses trabalhos são utilizadas diferentes abordagens, como

o uso de dados de redes sociais e shows. Em nossa pesquisa foram utilizados dados prévios do Top 50 e as características acústicas das músicas.

Realizamos as predições para dois meses após os dados de entrada. Utilizando um classificador SVM com *kernel* linear obtivemos acurácia, especificidade, revocação, valor-F1 e AUC acima de 70%. A utilização das características acústicas das faixas nos proveu resultados até 6,60% superiores em relação a quando não foram usadas.

Uma limitação do nosso trabalho é que só temos informações sobre as músicas enquanto elas aparecem no *ranking* “Top 50” do Spotify. Isso coincide com o nosso conceito de popularidade e, portanto, só podemos prever se uma música que *está* popular continuará sendo popular. Pretendemos, em nossos trabalhos futuros, contornar essa limitação expandindo o conjunto de músicas utilizando, por exemplo, *rankings* anteriores ao conjunto de treinamento como músicas não populares.

Em nossas pesquisas futuras devemos buscar uma forma de diminuir a quantidade de falsos positivos. Tal situação fez com que os valores de precisão e *fall-out* ficassem abaixo das outras métricas. Consideramos esse como o principal ponto a se desenvolver em nosso modelo. Uma abordagem para a diminuição dessa taxa de falsos positivos poderia ser utilizando de informações provenientes de redes sociais, que já tiveram sua importância no mercado musical demonstrada em outros estudos. Assim, pretendemos utilizar informações dessas redes com o intuito de obtermos melhores resultados.

## Referências

- Abel, F., Diaz-Aviles, E., Henze, N., Krause, D., and Siehndel, P. (2010). Analyzing the blogosphere for predicting the success of music and movie products. In *2010 International Conference on Advances in Social Networks Analysis and Mining*, pages 276–280.
- Arakelyan, S., Morstatter, F., Martin, M., Ferrara, E., and Galstyan, A. (2018). 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. ACM.
- Araújo, C. V., Neto, R. M., Nakamura, F. G., and Nakamura, E. F. (2017). 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. ACM.
- Brownlee, J. (2017). *Introduction to Time Series Forecasting with Python: How to Prepare Data and Develop Models to Predict the Future*. Jason Brownlee.
- Dhar, V. and Chang, E. A. (2009). Does chatter matter? the impact of user-generated content on music sales. *Journal of Interactive Marketing*, 23(4):300 – 307.
- Dubnov, S. (2004). Generalization of spectral flatness measure for non-gaussian linear processes. *IEEE Signal Processing Letters*, 11(8):698–701.
- Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861 – 874. ROC Analysis in Pattern Recognition.
- Harris, D. and Harris, S. (2010). *Digital design and computer architecture*. Morgan Kaufmann.

- Herremans, D., Martens, D., and Sørensen, K. (2014). Dance hit song prediction. *Journal of New Music Research*, 43(3):291–302.
- Interiano, M., Kazemi, K., Wang, L., Yang, J., Yu, Z., and Komarova, N. L. (2018). Musical trends and predictability of success in contemporary songs in and out of the top charts. *Royal Society Open Science*, 5(5):171274.
- Lee, J. and Lee, J. (2018). Music popularity: Metrics, characteristics, and audio-based prediction. *IEEE Transactions on Multimedia*, 20(11):3173–3182.
- McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., and Nieto, O. (2015). librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, pages 18–25.
- McKinney, W. (2010). Data structures for statistical computing in python. In van der Walt, S. and Millman, J., editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56.
- Olson, D. L. and Delen, D. (2008). *Advanced data mining techniques*. Springer Science & Business Media.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Shulman, B., Sharma, A., and Cosley, D. (2016). Predictability of popularity: Gaps between prediction and understanding. In *Tenth International AAAI Conference on Web and Social Media*, pages 348–357.
- Silva, M. O., Rocha, L. M., and Moro, M. M. (2019). Collaboration profiles and their impact on musical success. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC '19*, pages 2070–2077, New York, NY, USA. ACM.
- Steininger, D. M. and Gatzemeier, S. (2013). Using the wisdom of the crowd to predict popular music chart success. In *Proceedings of the 21st European Conference on Information Systems*, page 215.
- Zhang, H. (2004). The optimality of naive bayes. *AA*, 1(2):3.