

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data

Matheus Batista Abrantes

PREDIÇÃO DE “HITS” ATRAVÉS DO SPOTIFY

Belo Horizonte
2021

Matheus Batista Abrantes

PREDIÇÃO DE “HITS” ATRAVÉS DO SPOTIFY

Trabalho de Conclusão de Curso apresentado
ao Curso de Especialização em Ciência de
Dados e Big Data como requisito parcial à
obtenção do título de especialista.

Belo Horizonte
2021

SUMÁRIO

1. Introdução	4
1.1. Contextualização.....	4
1.2. O problema proposto.....	5
2. Coleta de Dados	6
2.1. Dataset do Spotify.....	6
2.2. Dataset do Last.fm	10
3. Processamento/Tratamento de Dados	11
3.1. Dataset do Spotify.....	11
3.2. Dataset do Last.fm	12
3.3. Join e Dataset final	13
4. Análise e Exploração dos Dados	13
5. Criação de Modelos de Machine Learning	18
5.1. RandomForest.....	18
5.2. XGBoost.....	24
6. Apresentação dos Resultados	29
7. Links	31
REFERÊNCIAS	33

1. Introdução

O trabalho visa à análise de músicas através do *dataset* do Spotify, no intuito de buscar *insights* que possam auxiliar músicos, gravadoras e a própria plataforma a predizer se uma música tem potencial de sucesso ou não. Para tal fim, o estudo foi estruturado em seis partes. Na introdução, será apresentada a contextualização do assunto tratado e o problema proposto. Na sequência, serão apresentadas as informações acerca da coleta dos dados (Capítulo 2), os passos realizados para o tratamento inicial, consistente em sua limpeza e estruturação (Capítulo 3), a análise e a exploração desses dados, com vistas à obtenção de informações estatísticas relevantes (Capítulo 4), os modelos de machine learning desenvolvidos (Capítulo 5) e por fim, no Capítulo 6 serão apresentados os resultados obtidos.

1.1. Contextualização

Na era de grande desenvolvimento do mundo digital, um dos temas que mais desperta interesse, ligado ao mundo dos conteúdos online, é o da música, onde (a exemplo do cinema) o streaming surgiu como um canal importante para consumo.

A plataforma mais proeminente é o Spotify, com uma participação de mercado maior do que a de varejistas ou estações de rádio na era digital (AGUIAR e WALDFOGEL, 2018).

Baseado nas afirmações de que há uma atração cada vez maior dos consumidores por serviços pagos de música (SUN, 2019) e há um interesse crescente em explorar estratégias de marketing no Spotify (PÉREZ-VERDEJO *et al*, 2021), foi pensado em como a ciência dos dados poderia atuar nesse ramo.

Se pudéssemos prever as vendas dos produtos antes de eles serem lançados no mercado, os negócios seriam fáceis (NIJKAMP, 2018), e como música também é um negócio bilionário e em crescimento (AL-BEITAWI *et al*, 2020), um modelo de machine learning que possa prever o sucesso de uma canção seria promissor.

A previsão de sucesso é útil para músicos, gravadoras e fornecedores de música porque as canções populares geram receitas maiores e permitem que os

artistas compartilhem sua mensagem com um público amplo (MIDDLEBROOK *et al*, 2019).

1.2. O problema proposto

A proposta desse trabalho consiste na criação de um modelo de machine learning em Python, que possa prever se uma música será um sucesso ou não. Para facilitar o entendimento do problema e da solução a ser proposta utilizamos a técnica do 5Ws, que consiste em responder as seguintes perguntas:

- **(Why?) Por que esse problema é importante?**

As gravadoras investem bilhões de dólares em novos talentos em todo o mundo a cada ano (HERREMANS *et al*, 2014), então obter insights sobre o que realmente faz uma canção de sucesso proporcionaria enormes benefícios para a indústria musical. Inclui a possibilidade de vender modelos de previsão para gravadoras e artistas (NIJKAMP, 2018), bem como usá-los para melhorar os serviços do streaming aos consumidores de música.

- **(Who?) De quem são os dados analisados?**

Os dados são da API oficial do Spotify, sendo disponibilizados no Kaggle por Farooq Ansari, onde integram dados característicos de mais de 40.000 músicas, incluindo faixas marcadas como hit ou não pelo autor, que podem ser usadas para fazer um modelo de classificação com finalidade de prever se uma determinada faixa será um “hit”. Para enriquecer os dados do Spotify, foi utilizado o *dataset* “Music artists popularity”, conjunto de dados que consiste em mais de 1,4 milhão de artistas musicais presentes no banco de dados MusicBrainz, com seus nomes, *tags*, popularidade (ouvintes), com base em dados extraídos do Last.fm.

- **(What?): Quais os objetivos e o que iremos analisar?**

Serão analisados os *datasets* das décadas de 60 à década de 10, com objetivo de buscar uma forma de prever se uma música será “hit” ou se irá fracassar.

- **(Where?): Trata dos aspectos geográficos e logísticos de sua análise.**

Os *datasets* utilizados apresentam um “viés geográfico”, pois o campo em que o autor marcou as canções como “hit” ou não, possui a condição de que o artista ou a música precisa ter aparecido, ao menos uma vez, na lista semanal da Billboard (conhecida como Hot 100), a qual é a principal forma de medir a popularidade dos artistas e músicas nos Estados Unidos.

(When?): Qual o período está sendo analisado?

Os *datasets* são dos anos de 1960 ao ano de 2019, totalizando 59 anos.

2. Coleta de Dados

Para esse trabalho, foram utilizados 2 *datasets* diferentes (Spotify¹ e Last.fm²), sendo que o *dataset* do Spotify está dividido em 6 arquivos, cada um com uma década específica (60, 70, 80, 90, 00 e 10). Todos os arquivos estão no formato CSV, foram disponibilizados no Kaggle e obtidos em 26/06/2021.

2.1. Dataset do Spotify

Os dados originais, incluem mais de 40.000 canções do mundo todo, com datas de lançamento entre 1960-2019. Cada faixa é identificada pelo nome da faixa, o nome do artista e um identificador de recurso exclusivo para a faixa, chamado de uri. O *dataset* foi disponibilizado no Kaggle em 25/04/2020 pelo usuário Farooq Ansari.

Os arquivos do Spotify (dataset-of-60s.csv, dataset-of-70s.csv, dataset-of-80s.csv, dataset-of-90s.csv, dataset-of-00s.csv e dataset-of-10s.csv) contêm a seguinte estrutura de variáveis:

Nome da coluna	Descrição	Tipo
track	Nome da canção.	object
artist	Nome(s) do(s) artista(s) relacionado(s) a canção.	object
uri	Código exclusivo que o Spotify usa para identificar a música.	object

¹ Base disponível em: <https://www.kaggle.com/theoverman/the-spotify-hit-predictor-dataset>

² Base disponível em: <https://www.kaggle.com/pieca111/music-artists-popularity>

key	A chave em que está a faixa. Os inteiros mapeiam os tons usando a notação padrão de classe de tom. Por exemplo. 0 = C, 1 = C# / D b, 2 = D e assim por diante.	integer
mode	Mode indica a modalidade (maior ou menor) de uma faixa, o tipo de escala da qual seu conteúdo melódico é derivado. O maior é representado por 1 e o menor é 0.	integer
time_signature	Uma estimativa de fórmula de compasso geral de uma faixa. A fórmula de compasso (medidor) é uma convenção notacional para especificar quantas batidas existem em cada barra (ou compasso).	integer
danceability	A capacidade de dança descreve como uma faixa é adequada para dançar com base em uma combinação de elementos musicais, incluindo andamento, estabilidade do ritmo, intensidade da batida e regularidade geral. Um valor de 0,0 é menos dançante e 1,0 é mais dançante.	float
energy	Energy é uma medida de 0,0 a 1,0 e representa uma medida perceptual de intensidade e atividade. Normalmente, as faixas energéticas parecem rápidas, altas e barulhentas. Por exemplo, death metal tem alta energia, enquanto um prelúdio de Bach tem pontuação baixa na escala.	float
loudness	O volume geral de uma faixa em decibéis (dB). Os valores de sonoridade são calculados em toda a trilha e são úteis para comparar a sonoridade relativa das trilhas.	float
speechiness	Speechiness detecta a presença de palavras faladas em uma faixa. Quanto mais exclusivamente falada for a gravação (por exemplo, talk show, audiolivro, poesia), mais próximo de 1,0 será o valor do atributo. Valores entre 0,33 e 0,66 descrevem faixas que podem conter música e fala.	float
acousticness	Uma medida de confiança de 0,0 a 1,0 para saber se a faixa é acústica. 1.0 representa alta confiança de que a faixa é acústica.	float

instrumentalness	Prevê se uma faixa não contém vocais. Os sons “Ooh” e “aah” são tratados como instrumentais neste contexto. Faixas de rap ou palavra falada são claramente “vocais”. Quanto mais próximo o valor da instrumentalidade estiver de 1,0, maior será a probabilidade de a faixa não conter conteúdo vocal.	float
liveness	Detecta a presença de um público na gravação. Valores de vivacidade mais altos representam um aumento na probabilidade de a trilha ser executada ao vivo. Um valor acima de 0,8 fornece uma grande probabilidade de que a faixa esteja ao vivo.	float
valence	Uma medida de 0,0 a 1,0 que descreve a positividade musical transmitida por uma faixa. Faixas com alta valência soam mais positivas (por exemplo, feliz, alegre, eufórico), enquanto faixas com baixa valência soam mais negativas (por exemplo, triste, deprimido, com raiva).	float
tempo	O tempo estimado geral de uma faixa em batidas por minuto (BPM).	float
duration ms	A duração da faixa em milissegundos.	integer
chorus_hit	Uma estimativa de quando o refrão da faixa aparece pela primeira vez.	float
sections	Número de seções dentro da música.	integer
target	Uma variável booleana que descreve se a faixa já apareceu na lista Weekly Hot-100 da Billboard.	boolean

Sobre a coluna *target*, o autor definiu as seguintes condições para classificar uma faixa como *hit* (*target* = 1):

- A faixa deve constar da lista de *hits* daquela década.
- O artista da faixa deve constar da lista de *hits* da década.
- A faixa deve pertencer a um gênero que possa ser considerado *mainstream*.
- O gênero da faixa deve ter uma música na lista de *hits*.
- A faixa deve ter os Estados Unidos como um de seus mercados.

A tabela abaixo mostra quantas variáveis cada dataset do Spotify possui, como também a quantidade de registros em cada uma:

Nome do arquivo	Quantidade de variáveis	Quantidade de registros
dataset-of-60s.csv	19	8.642
dataset-of-70s.csv	19	7.766
dataset-of-80s.csv	19	6.908
dataset-of-90s.csv	19	5.520
dataset-of-00s.csv	19	5.872
dataset-of-10s.csv	19	6.398

A soma dos registros nos *datasets* do Spotify totaliza 41.106 músicas. Na figura 1, são apontadas as variáveis e a quantidade de valores nulos.

	Tipo	Nulos	Nulos (%)
track	object	0	0.0
artist	object	0	0.0
uri	object	0	0.0
danceability	float64	0	0.0
energy	float64	0	0.0
key	int64	0	0.0
loudness	float64	0	0.0
mode	int64	0	0.0
speechiness	float64	0	0.0
acousticness	float64	0	0.0
instrumentalness	float64	0	0.0
liveness	float64	0	0.0
valence	float64	0	0.0
tempo	float64	0	0.0
duration_ms	int64	0	0.0
time_signature	int64	0	0.0
chorus_hit	float64	0	0.0
sections	int64	0	0.0
target	int64	0	0.0

Figura 1 – Análise de valores nulos na base Spotify

Fonte: Autoria própria

Nesse cenário, não foram detectados valores nulos nos *datasets* do Spotify.

2.2. Dataset do Last.fm

Os dados originais, incluem mais de 1,4 milhão de perfis de artistas com dados disponíveis no site do Last.fm e na base de dados do MusicBrainz. O *dataset* foi disponibilizado no Kaggle em 05/05/2019 pelo usuário Piotr.

O arquivo do Last.fm (lastfm.csv) contém a seguinte estrutura de variáveis:

Nome da coluna	Descrição	Tipo
mbid	Código único para cada artista.	object
artist_mb	Nome do artista na base MusicBrainz.	object
artist_lastfm	Nome do artista na base Last.fm.	object
country_mb	País do artista na base MusicBrainz.	object
country_lastfm	País do artista na base Last.fm.	object
tags_mb	Gêneros relacionados ao artista na base MusicBrainz.	object
tags_lastfm	Mostra o gênero relacionado ao artista no Last.fm, como também mostra as tags geradas pelos usuários do site.	object
listeners_lastfm	Quantidade de ouvintes no Last.fm.	float
scrobbles_lastfm	Quantidade de vezes que ao menos uma música daquele artista foi tocada.	float
ambiguous_artist	Booleano, que mostra se o artista possui mais de uma conta no site do Last.fm.	boolean

A tabela abaixo mostra quantas variáveis cada *dataset* do Spotify possui, como também a quantidade de registros em cada uma:

Nome do arquivo	Quantidade de variáveis	Quantidade de registros
lastfm.csv	10	1.466.083

Na figura 2, são apontadas as variáveis e a quantidade de valores nulos.

	Tipo	Nulos	Nulos (%)
mbid	object	0	0.0
artist_mb	object	8	0.000546
artist_lastfm	object	479327	32.694397
country_mb	object	803715	54.820566
country_lastfm	object	1254585	85.573941
tags_mb	object	1346137	91.818608
tags_lastfm	object	1085008	74.00727
listeners_lastfm	float64	479323	32.694124
scrobbles_lastfm	float64	479323	32.694124
ambiguous_artist	bool	0	0.0

Figura 2 – Análise de valores nulos na base Last.fm

Fonte: Autoria própria

Fica evidente que deverá ser feito um tratamento para os valores nulos, pois, pegando como exemplo a coluna *tags_mb*, a quantidade é enorme. O tratamento será feito no próximo capítulo.

3. Processamento/Tratamento de Dados

Para processamento, tratamento e análise de dados foi utilizada a linguagem Python, via Jupyter Notebooks para facilitar o acompanhamento e reprodução do trabalho realizado.

De início, foram buscados possíveis problemas e soluções sobre os *datasets*, embasando-se em suas estruturas e tipos de dados.

3.1. Dataset do Spotify

Baseando-se na coluna *artist*, é possível prever que haverá um problema futuro quando for feito o *join* entre as bases Spotify com a base Last.fm, pois nessa coluna estão registrados os artistas relacionados, sendo possível ter mais de um artista por música, ao contrário da base Last.fm, que possui apenas um artista por registro.

Decidiu-se inserir uma nova coluna, chamada *artist_name*, onde será retirada a palavra *Featuring* (que indica uma participação) e qualquer caractere posterior. Como os dados do Spotify estão em 6 arquivos, a inserção foi feita através de um loop. Mais ajustes foram feitos na coluna *artist_name*, como: retirar espaços, vírgulas, barras, X (devido algumas músicas utilizarem o X como sinônimo para *Featuring*) e seus caracteres posteriores no fim da *string*, além de corrigir o nome da artista Beyoncé.

Os *datasets* do Spotify não foram unificados, com objetivo de manter uma maior flexibilidade na análise e, principalmente, nos testes dos modelos de machine learning.

3.2. Dataset do Last.fm

Embasando-se nas colunas do *dataset* do Last.fm, foram escolhidas as seguintes colunas para gerar o *dataset* final: *artist_mb*, *country_mb* e *tags_lastfm* (renomeadas para *artist_final*, *country* e *genre* respectivamente). Sobre a coluna *genre*, foi feito um tratamento para pegar apenas o primeiro valor antes da vírgula, que representa a *tag* mais utilizada para descrever o estilo de música do artista.

Na figura 3, são mostradas as colunas, seus tipos e a quantidade de valores nulos.

	Tipo	Nulos	Nulos (%)
artist_final	object	8	0.000546
country	object	803715	54.820566
genre	object	1085008	74.00727

Figura 3 – Análise de valores nulos na base Last.fm

Fonte: Autoria própria

Essas colunas foram as escolhidas para a base final devido terem uma taxa menor de valores nulos quando comparadas com as colunas *country_lastfm* (85,57%) e *tags_mb* (91,81%).

Apesar da base original do Last.fm possuir 1.466.083 registros, a coluna *artist_name* possui 1.352.997 registros únicos, ou seja, além da questão dos valores nulos, também há o tema sobre os registros duplicados. Decidiu-se pela exclusão

das linhas com valores nulos, pois devido a coluna *genre* (gênero do artista) possuir um altíssimo percentual de nulidade e o gênero ser um dado que não possui média, a manutenção dos valores poderia comprometer a análise por gênero musical. Com a exclusão das linhas com valores nulos, o *dataset* ficou com 194.409 registros, sendo que 180.961 são únicos.

Os valores duplicados também foram excluídos, para evitar um produto cartesiano quando o *join* for realizado. Foi mantida apenas a primeira ocorrência de cada registro, concluindo a base final do Last.fm em 180.961 registros. Comparando as bases iniciais e finais, houve uma perda de 87,65% (1.285.122 registros).

3.3. Join e Dataset final

Com objetivo de enriquecer as informações contidas nas bases do Spotify, foi realizado o *join* entre os *datasets* finais do Spotify com o *dataset* final do Last.fm.

Foram utilizadas as colunas *artist_name* (Spotify) e *artist_final* (Last.fm) para realização do *left join*.

Os arquivos finais do Spotify ficaram com as seguintes estruturas e quantidade de registros:

Nome do arquivo	Quantidade de variáveis	Quantidade de registros
FINAL_dataset-of-60s.csv	23	7.287
FINAL_dataset-of-70s.csv	23	6.470
FINAL_dataset-of-80s.csv	23	5.733
FINAL_dataset-of-90s.csv	23	4.347
FINAL_dataset-of-00s.csv	23	4.830
FINAL_dataset-of-10s.csv	23	4.389

Os registros finais totalizam 33.056 músicas. Comparando com as bases originais, houve uma perda de 8.050 músicas (19,58%).

4. Análise e Exploração dos Dados

Para essa etapa, criou-se um *dataframe* abrangendo todos os *dataset* Spotify, com intuito de realizar uma análise geral dos dados. O *dataset* final para análise é composto por 23 colunas e 33.056 registros.

Desenvolveu-se uma matriz de correlação (figura 4), onde alguns *insights* podem ser obtidos:

- Não há correlação entre a variável *energy* com *acousticness*, o que é compreensível, pois, no geral, músicas mais enérgicas não são acústicas.
- Analisando a variável *target* (a qual é chave desse trabalho), compreende-se que músicas instrumentais e músicas acústicas não se tornam *hits* de sucesso, pois dispõe de uma correlação baixíssima.
- Por outro lado, quando analisadas as variáveis *danceability*, *energy*, *loudness* e *valence*, nota-se que há correlação, o que significa que músicas dançantes, altas e positivas possuem grande probabilidade de se tornarem um sucesso.

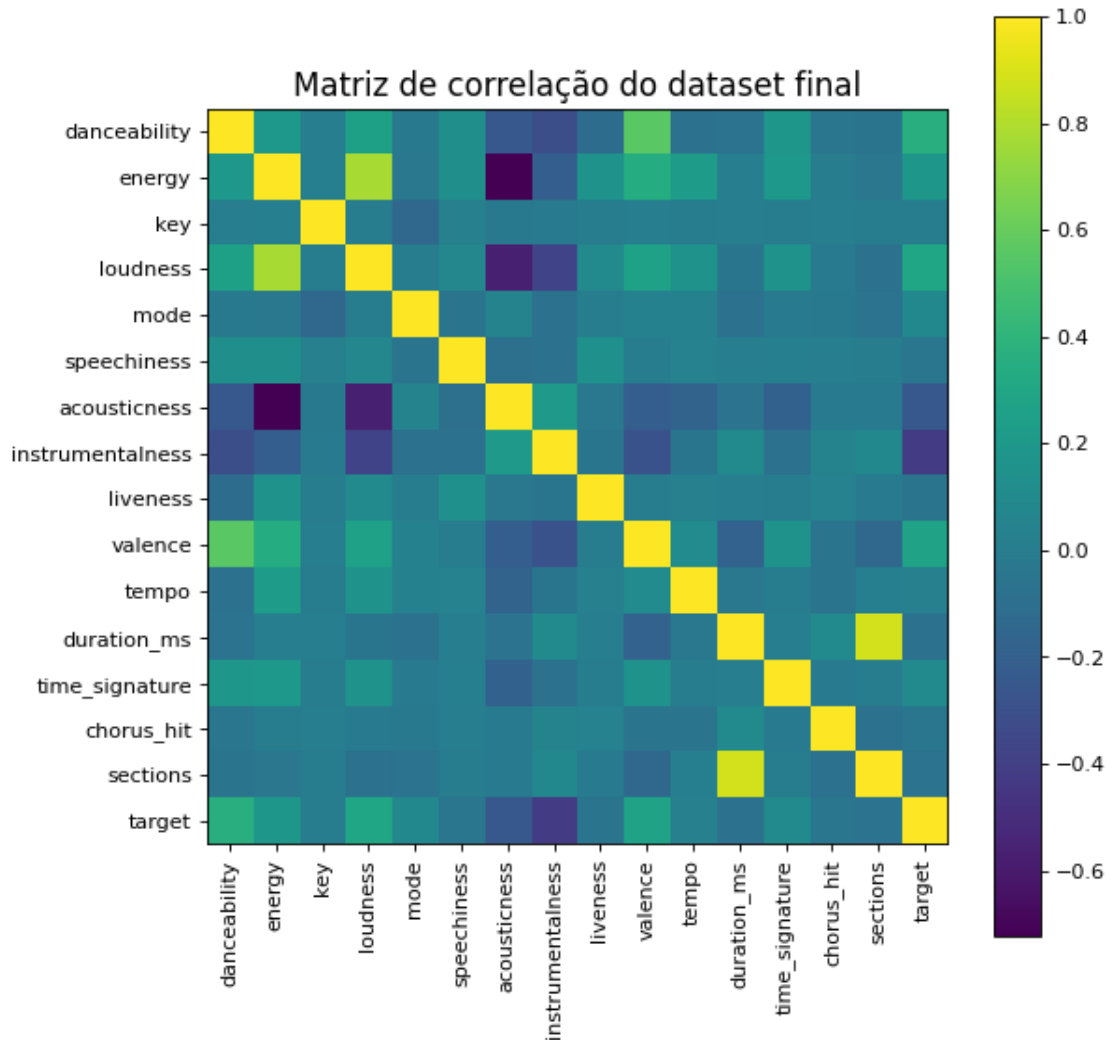


Figura 4 – Matriz de correlação do dataset Spotify

Fonte: Autoria própria

A figura 5 exibe o top 5 dos países com mais *hits* no *dataset*, onde é possível entender graficamente a questão do “viés geográfico”, pois os Estados Unidos possuem uma quantidade maior do que a de todos os outros participantes do top 5 somados.

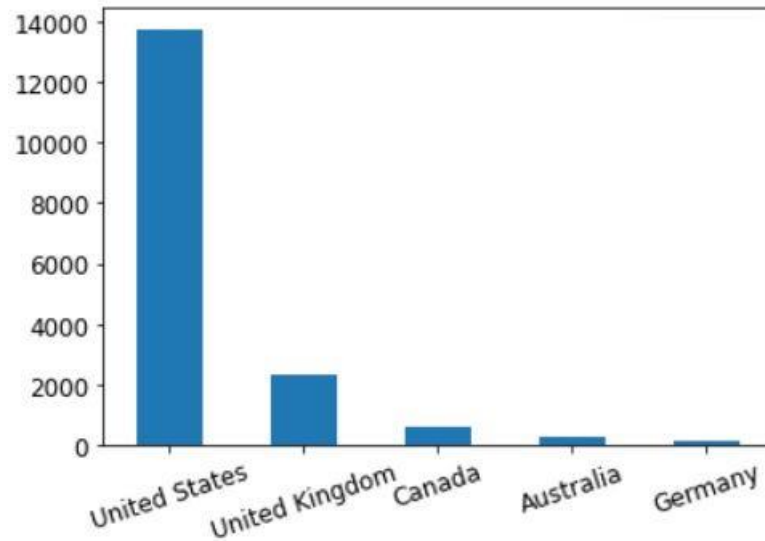


Figura 5 – Análise dos top 5 países no dataset Spotify

Fonte: Autoria própria

No seguinte gráfico (figura 6), pode-se observar o top 5 dos artistas com mais *hits* no *dataset*. Interessante notar que do top 5, apenas 3 são norte-americanos (Glee Cast, Taylor Swift, The Beatles), porém todos têm em comum o idioma inglês.

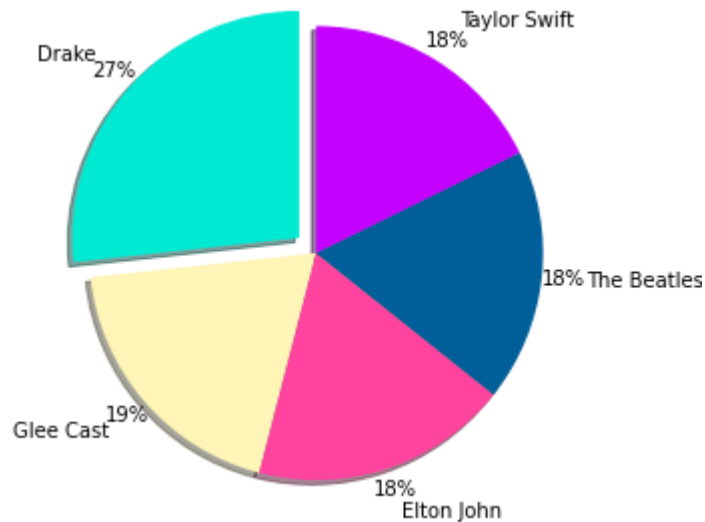


Figura 6 – Análise dos top 5 artistas no dataset Spotify

Fonte: Autoria própria

Desenvolveu-se uma nuvem de palavras (figura 7) no formato do logotipo do Spotify. Foi utilizado um novo *dataframe*, contando apenas com registros na coluna *target* marcados com 1, ou seja, apenas registros que são considerados *hit*.

Foram obtidos alguns insights ao analisar os gêneros musicais:

- Nota-se que alguns gêneros foram perdidos, pois ficaram definidos apenas com o número da década ou definidos como *oldies*.
- O soul foi um gênero de grande relevância nas décadas de 60, 70 e 80, porém não apareceu mais entre os top 5 nos últimos 30 anos analisados.
- Rock (6x) e pop (5x) são os gêneros com mais aparições.
- O rock, porém, está enfraquecendo pois na última década possui o pior número entre todas as décadas e gêneros.

5. Criação de Modelos de Machine Learning

Para escolha dos modelos utilizados, realizou-se pesquisas na literatura relacionadas ao machine learning aplicado aos dados musicais. O XGBoost foi um dos escolhidos, devido ter sido considerado o melhor classificador baseado em recursos relativos à música (BAHULEYAN, 2018). Para comparação de resultados, foi selecionado também o RandomForest, devido ter obtido um ótimo resultado em um estudo sobre o sucesso das músicas no Reino Unido (INTERIANO *et al*, 2018) e em outro estudo sobre classificação de gêneros musicais na Espanha (LÓPEZ *et al*, 2017).

5.1. RandomForest

O desenvolvimento dos modelos em machine learning foi iniciado pelo RandomForest. RandomForest é um método preditivo baseado em árvores de classificação, que são conhecidas como uma ferramenta preditiva flexível para modelar funções lineares e não lineares de recursos (INTERIANO *et al*, 2018). Foi realizado um loop para selecionar os seis *datasets* do Spotify e depois unificá-los (figura 9).

```

1 # Loop para pegar todos os CSV
2
3 extensao = 'csv'
4 arquivos = [i for i in glob.glob('*.{}'.format(extensao))]
5
6 arquivos

Pressione Ctrl+Alt+Enter para executar a célula

['FINAL_dataset-of-00s.csv',
 'FINAL_dataset-of-10s.csv',
 'FINAL_dataset-of-60s.csv',
 'FINAL_dataset-of-70s.csv',
 'FINAL_dataset-of-80s.csv',
 'FINAL_dataset-of-90s.csv']

1 # Juntando os arquivos em um só
2
3 data = pd.concat([pd.read_csv(f) for f in arquivos ])
4 data.reset_index(drop=True, inplace=True)

```

Figura 9 – Loop para gerar dataframe

Fonte: Autoria própria

No passo seguinte, realizou-se a seleção das colunas que serviram de parâmetro ao modelo e a divisão da base entre treino e teste (figura 10).

```

1 # Selecionando as colunas que servirão como parâmetro no modelo
2
3 y = data.target
4 x = data[['danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness', 'acousticness',
5 | | | 'instrumentalness', 'liveness', 'valence', 'tempo', 'duration_ms', 'time_signature', 'chorus_hit', 'sections']]

+ Código + Markdown

1 # Total de linhas
2
3 print(x.shape, y.shape)

(33056, 15) (33056,)

1 # Divisão da base em treino e teste
2
3 x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=1/3, random_state=5)

```

Figura 10 – Seleção de colunas e divisão da base

Fonte: Autoria própria

Na figura 11, temos a criação do classificador RandomForest, onde o número de estimadores foi fixado em 500, visando a qualidade na performance e baseando-se no estudo realizado por Oshiro em 2012.

RANDOM FOREST

```

1 # Criando um classificador RandomForest
2
3 modelo = RandomForestClassifier(n_estimators=500, criterion='gini', random_state = 0)
4 modelo = modelo.fit(x_train, y_train)
✓ 21.7s

```

Figura 11 – Criação do modelo RandomForest

Fonte: Autoria própria

Na figura 12, realizou-se a criação da matriz de confusão do modelo, onde foi utilizada a palavra “*flop*” como antônimo de *hit*, baseando-se em como é chamada uma música considerada um fracasso pelo autor do *dataset*.

```

1 # Matriz de confusão
2
3 y_pred = modelo.predict(x_test)
4
5 cf_matrix = confusion_matrix(y_test, y_pred)
6
7 print(classification_report(y_test, y_pred))
8
9 ax= plt.subplot()
10 sns.heatmap(cf_matrix, annot=True, fmt='g', ax=ax);
11
12 # labels, title and ticks
13 ax.set_xlabel('Previsões do RandomForest');ax.set_ylabel('Valores reais');
14 ax.set_title('Matriz de Confusão');
15 ax.xaxis.set_ticklabels(['Hit', 'Flop']); ax.yaxis.set_ticklabels(['Hit', 'Flop']);
✓ 2.2s

```

Figura 12 – Criação da matriz de confusão

Fonte: Autoria própria

A figura 13 apresenta o resultado do modelo, através da matriz de confusão:

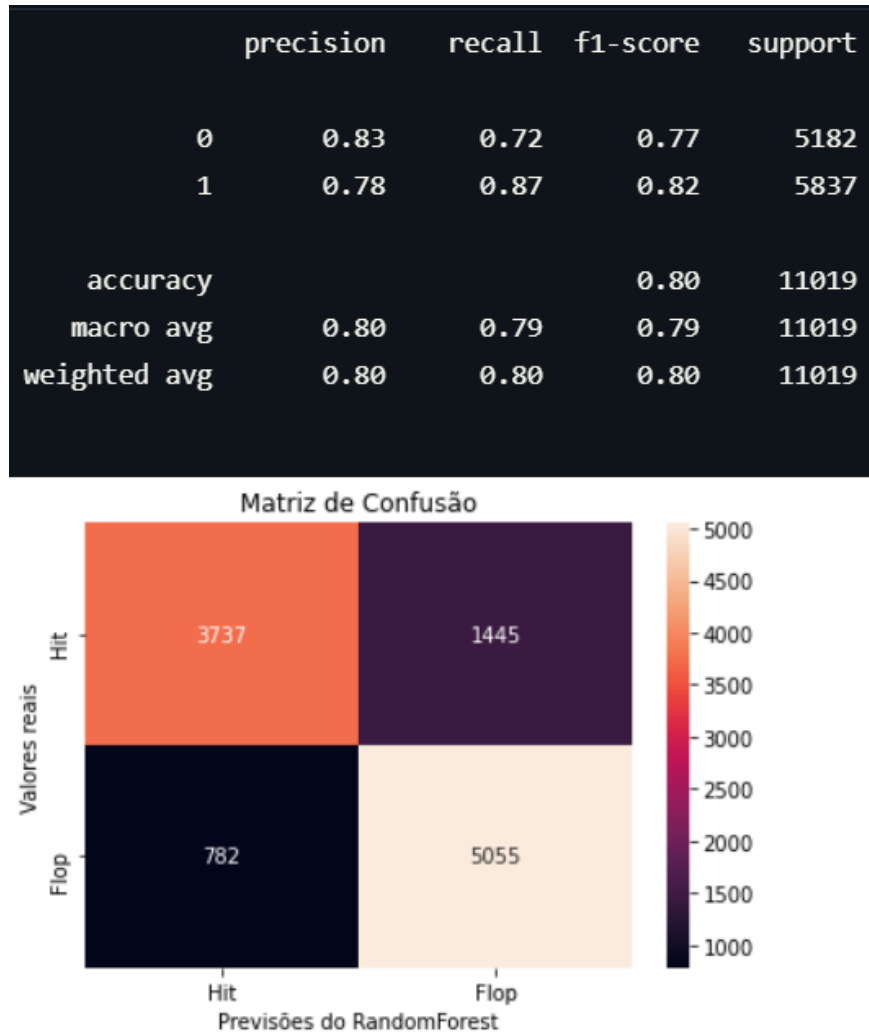


Figura 13 – Resultado do modelo RandomForest

Fonte: Autoria própria

Como resultado, obteve-se 80% de acurácia (quanto o modelo acertou das previsões), recall (quão bom o modelo é para prever positivos) de 87% em prever um *hit*, precisão (quão bem o modelo trabalhou) de 78% e f1-score (balanço entre a precisão e o recall) de 82%.

Na figura 14, foram classificados os parâmetros utilizados por sua importância no modelo. Individualmente, cada recurso foi bastante fraco como um indicador de sucesso ou não de uma música, sendo assim, todos os parâmetros são importantes para o sucesso do modelo.

```

1 # Classificando os parâmetros por sua importância no modelo
2
3 sorted(zip(modelo.feature_importances_, x.columns), reverse=True)
✓ 0.1s
(0.1700374700491776, 'instrumentalness'),
(0.11077487573003265, 'acousticness'),
(0.1083704030976634, 'danceability'),
(0.08358085994582654, 'energy'),
(0.07567285806480462, 'duration_ms'),
(0.07507568782139426, 'loudness'),
(0.0724863592480964, 'speechiness'),
(0.07033322474719204, 'valence'),
(0.05094981089852363, 'tempo'),
(0.05046265870341049, 'liveness'),
(0.0485425202772903, 'chorus_hit'),
(0.03854214130915587, 'sections'),
(0.028620249592750474, 'key'),
(0.01015300956621256, 'mode'),
(0.006397870948469031, 'time_signature')]

```

Figura 14 – Classificação dos parâmetros

Fonte: Autoria própria

Visando a possibilidade de melhora de resultado, foi desenvolvido um novo teste, abrangendo apenas os *datasets* das décadas de 00 e 10. Conforme visto na análise dos dados, os gostos musicais mudaram bastante e talvez limitar os dados a um bloco de vinte anos melhoraria a precisão, podendo prever com maior assertividade o que seria considerado um sucesso hoje.

Na figura 15, obteve-se o resultado dos novos testes, o modelo atingiu 87% de acurácia, recall de 92% em prever um hit, precisão de 86% e f1-score de 88%. Em resumo, a precisão do modelo de fato melhorou.

	precision	recall	f1-score	support
0	0.89	0.81	0.85	1384
1	0.86	0.92	0.88	1689
accuracy			0.87	3073
macro avg	0.87	0.86	0.87	3073
weighted avg	0.87	0.87	0.87	3073

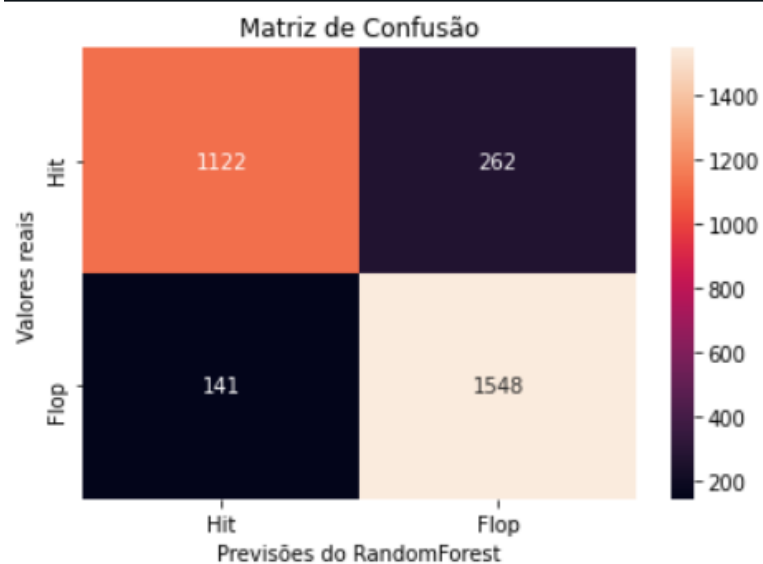


Figura 15 – Resultado do modelo RandomForest

Fonte: Autoria própria

Na figura 16, novamente cada recurso foi bastante fraco como um indicador de sucesso ou não de uma música, tendo apenas um aumento na variável *instrumentalness*.

```
(0.2567008637368943, 'instrumentalness'),
(0.11736678452141436, 'danceability'),
(0.09731250401431281, 'loudness'),
(0.09246240202395288, 'acousticness'),
(0.08697532736405224, 'duration_ms'),
(0.07936670676598916, 'energy'),
(0.05908558762752952, 'valence'),
(0.042887179476228506, 'speechiness'),
(0.03850325121179015, 'liveness'),
(0.035289911772094194, 'tempo'),
(0.03505192299839428, 'chorus_hit'),
(0.027241710261753385, 'sections'),
(0.021127300722229705, 'key'),
(0.0059749011482830286, 'mode'),
(0.0046536463550815475, 'time_signature')]
```

Figura 16 – Classificação dos parâmetros

Fonte: Autoria própria

5.2. XGBoost

O segundo e último algoritmo de machine learning a ser desenvolvido foi o XGBoost. A exemplo do RandomForest, também foi realizado um loop para selecionar os seis *datasets* do Spotify e depois unificá-los (figura 17).

```
1 # Loop para pegar todos os CSV
2
3 extensao = 'csv'
4 arquivos = [i for i in glob.glob('*.{}'.format(extensao))]
5
6 arquivos

Pressione Ctrl+Alt+Enter para executar a célula

['FINAL_dataset-of-00s.csv',
 'FINAL_dataset-of-10s.csv',
 'FINAL_dataset-of-60s.csv',
 'FINAL_dataset-of-70s.csv',
 'FINAL_dataset-of-80s.csv',
 'FINAL_dataset-of-90s.csv']

1 # Juntando os arquivos em um só
2
3 data = pd.concat([pd.read_csv(f) for f in arquivos ])
4 data.reset_index(drop=True, inplace=True)
```

Figura 17 – Loop para gerar dataframe

Fonte: Autoria própria

No passo seguinte, foram selecionadas as mesmas colunas que serviram de parâmetro ao modelo RandomForest (figura 18).



```

1 # Selecionando as colunas que servirão como parâmetro no modelo
2
3 y = data.target
4 x = data[['danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness', 'acousticness',
5 | | | 'instrumentalness', 'liveness', 'valence', 'tempo', 'duration_ms', 'time_signature', 'chorus_hit', 'sections']]

```

+ Código + Markdown

```

1 # Total de linhas
2
3 print(x.shape, y.shape)

```

(33056, 15) (33056,)

```

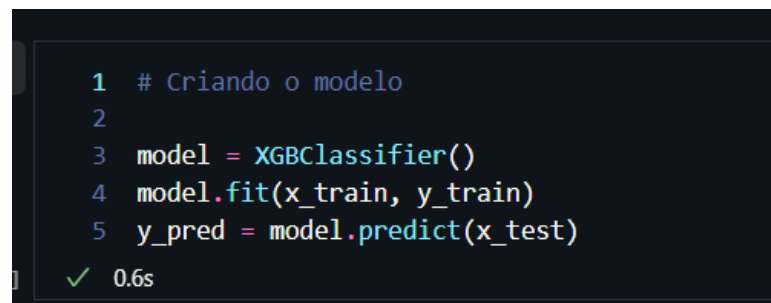
1 # Divisão da base em treino e teste
2
3 x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=1/3, random_state=5)

```

Figura 18 – Seleção de colunas e divisão da base

Fonte: Autoria própria

Na figura 19, temos a criação do classificador XGBoost.



```

1 # Criando o modelo
2
3 model = XGBClassifier()
4 model.fit(x_train, y_train)
5 y_pred = model.predict(x_test)

```

✓ 0.6s

Figura 19 – Criação do modelo XGBoost

Fonte: Autoria própria

Na figura 20, realizou-se a criação da matriz de confusão do modelo, utilizando as mesmas propriedades da matriz de confusão do modelo RandomForest.

```

1 # Matriz de confusão
2
3 y_pred = model.predict(x_test)
4
5 cf_matrix = confusion_matrix(y_test, y_pred)
6
7 print(classification_report(y_test, y_pred))
8
9 ax= plt.subplot()
10 sns.heatmap(cf_matrix, annot=True, fmt='g', ax=ax);
11
12 # labels, title and ticks
13 ax.set_xlabel('Previsões do XGBoost');ax.set_ylabel('Valores reais');
14 ax.set_title('Matriz de Confusão');
15 ax.xaxis.set_ticklabels(['Hit', 'Flop']); ax.yaxis.set_ticklabels(['Hit', 'Flop']);

```

✓ 0.2s

Figura 20 – Criação da matriz de confusão

Fonte: Autoria própria

A figura 21 apresenta o resultado do modelo, através da matriz de confusão:

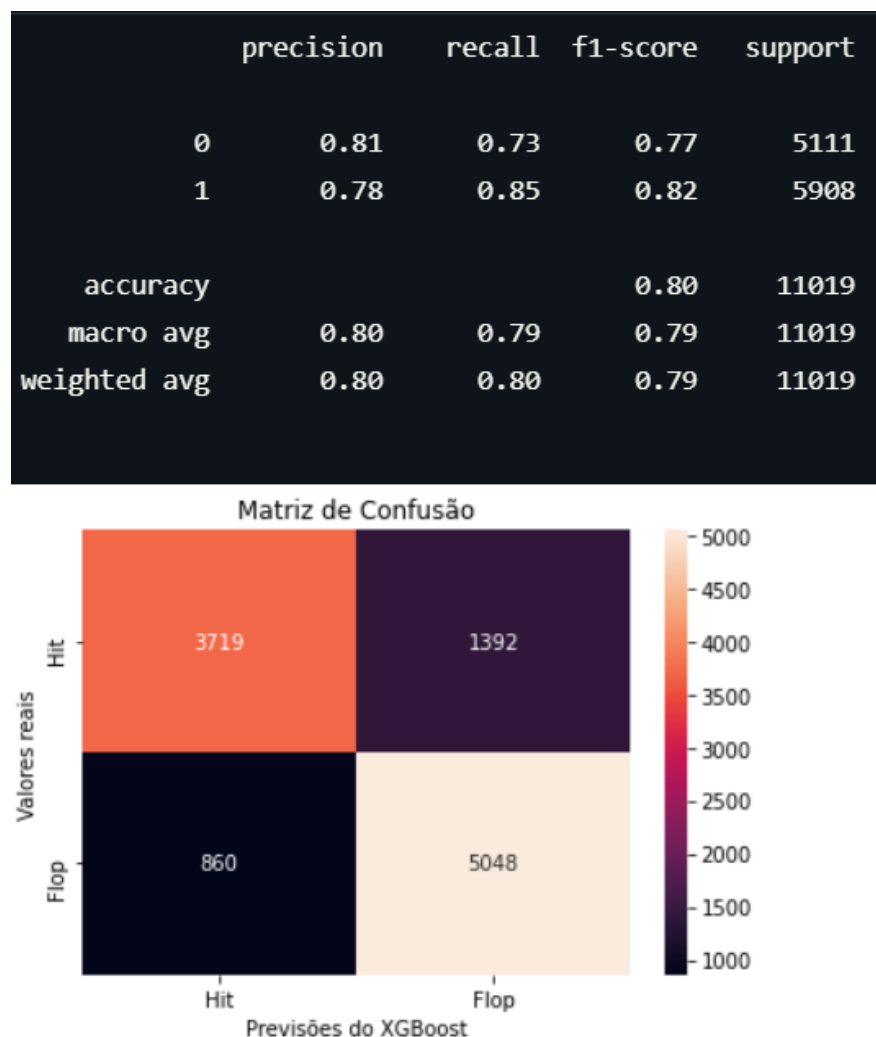


Figura 21 – Resultado do modelo XGBoost

Fonte: Autoria própria

Como resultado, obteve-se 80% de acurácia, recall de 85% em prever um *hit*, precisão de 78% e f1-score de 82%.

Na figura 22, foram classificados os parâmetros utilizados por sua importância no modelo. Novamente, cada recurso foi bastante fraco como um indicador de sucesso ou não de uma música e o *instrumentalness* teve uma importância maior para esse modelo, quando comparado ao modelo RandomForest.

```
(0.29219124, 'instrumentalness'),
(0.09378996, 'acousticness'),
(0.0839424, 'danceability'),
(0.077934295, 'mode'),
(0.060282163, 'duration_ms'),
(0.055350818, 'energy'),
(0.054587144, 'speechiness'),
(0.044641756, 'time_signature'),
(0.04449601, 'valence'),
(0.03879705, 'loudness'),
(0.037995804, 'sections'),
(0.03472083, 'tempo'),
(0.028903615, 'chorus_hit'),
(0.027756361, 'liveness'),
(0.02461059, 'key')]
```

Figura 22 – Classificação dos parâmetros

Fonte: Autoria própria

Assim como com o modelo RandomForest, visando a possibilidade de melhora de resultado, foi desenvolvido um novo teste, abrangendo apenas os *datasets* das décadas de 00 e 10.

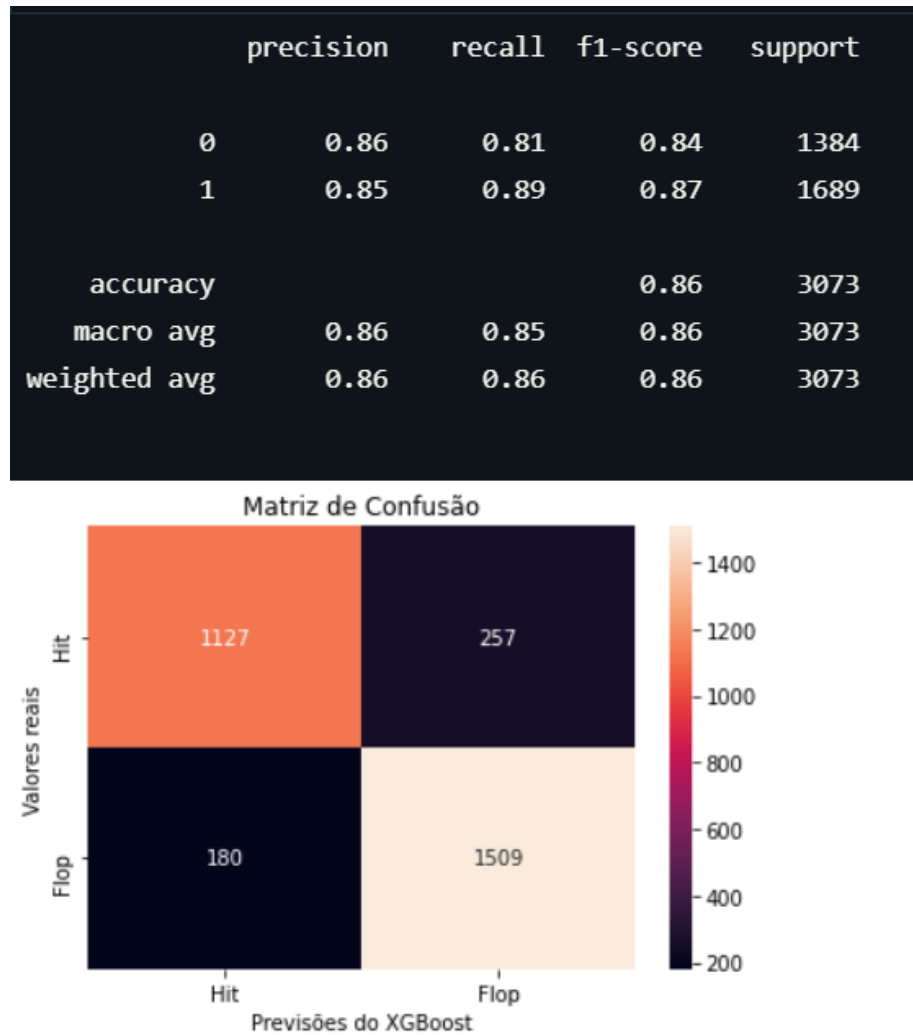


Figura 23 – Resultado do modelo XGBoost

Fonte: Autoria própria

Na figura 23, obteve-se o resultado dos novos testes, o modelo atingiu 86% de acurácia, recall de 89% em prever um hit, precisão de 85% e f1-score de 87%. Novamente, houve melhora na precisão do modelo.

```
(0.41943157, 'instrumentalness'),
(0.076925464, 'danceability'),
(0.06097241, 'loudness'),
(0.060014863, 'energy'),
(0.05643414, 'duration_ms'),
(0.047765303, 'acousticness'),
(0.045431104, 'valence'),
(0.035584893, 'time_signature'),
(0.030996516, 'liveness'),
(0.029786607, 'sections'),
(0.029398823, 'speechiness'),
(0.028665928, 'tempo'),
(0.027220245, 'chorus_hit'),
(0.025936557, 'key'),
(0.025435513, 'mode')]
```

Figura 24 – Classificação dos parâmetros

Fonte: Autoria própria

Na figura 24, a variável *instrumentalness* teve um percentual alto de importância quando comparada com os 25,67% atingidos no segundo teste do RandomForest.

6. Apresentação dos Resultados

Foi elaborado um workflow com as etapas pelas quais o trabalho foi desenvolvido, com intuito de representar os processos:

Título: Predição de “hits” através do Spotify		
Declaração do problema:	Resultados e previsões:	Aquisição de dados:
Falta de insights sobre o que realmente faz uma canção de sucesso.	As previsões têm como objetivo prever se uma música será um <i>hit</i> ou não.	Os dados foram coletados através do Kaggle (dataset Spotify e Last.fm).
Modelagem:	Avaliação do modelo:	Preparação dos dados:
Análise dos Dados em Python, com enriquecimento dos datasets com o propósito de criar modelos em machine learning.	A avaliação dos modelos foi feita através da matriz de confusão, tendo obtido bons resultados.	Dados duplicados e nulos foram retirados da base final, além disso foi tratada a coluna com nome dos artistas.

Foi elaborado um infográfico com intuito de apresentar o trabalho de forma mais simples e que possa ser compreendido pelo maior número de pessoas, sendo elas especialistas na área ou não.

Como forma de agregar conteúdo, foi desenvolvido um dashboard em Power BI, para proporcionar a visualização das seis principais variáveis através de gráficos, bem como os artistas com mais “*hits*” no *dataset* e como estão distribuídos os “*hits*” no mapa global.

POR MATHEUS ABRANTES



2:49

4:55



O que torna uma música em um "hit" no Spotify ?



Foram analisados dados das últimas 6 décadas do Spotify e alguns dados do Last.fm



Foi feita uma limpeza e união dessas bases usando a linguagem Python



Um algoritmo foi criado, para o sistema aprender com as músicas e prever o "hit"



O algoritmo atingiu 87% de
acurácia ao analisar as
músicas

Predição de hits



Por Matheus Abrantes



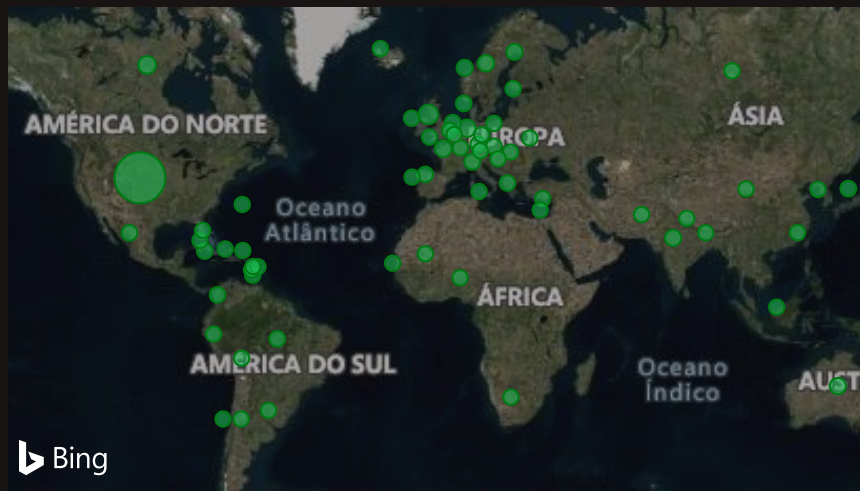


MÚSICAS
● HIT ● FLOP

DÉCADAS
00 10

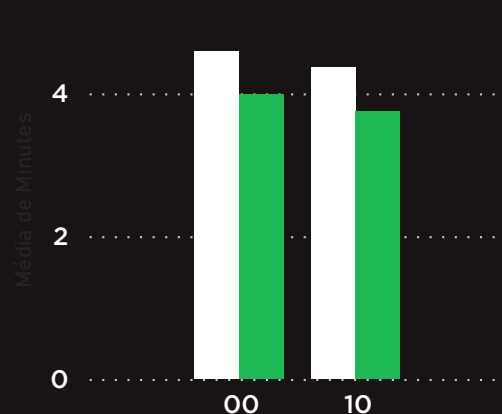


MÚSICAS POR PAÍS

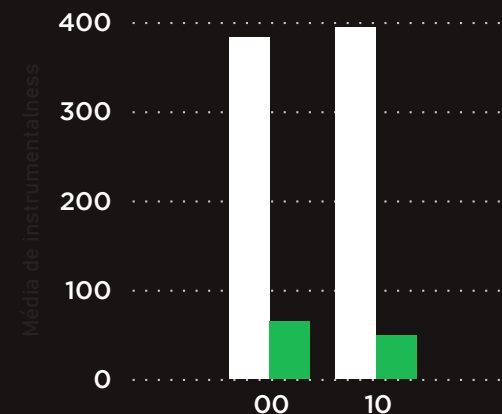


MÉDIAS DAS PRINCIPAIS VARIÁVEIS

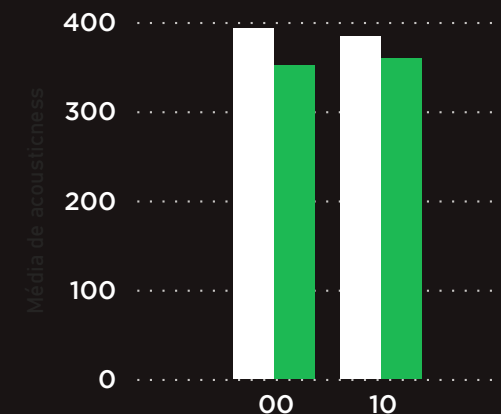
MINUTOS



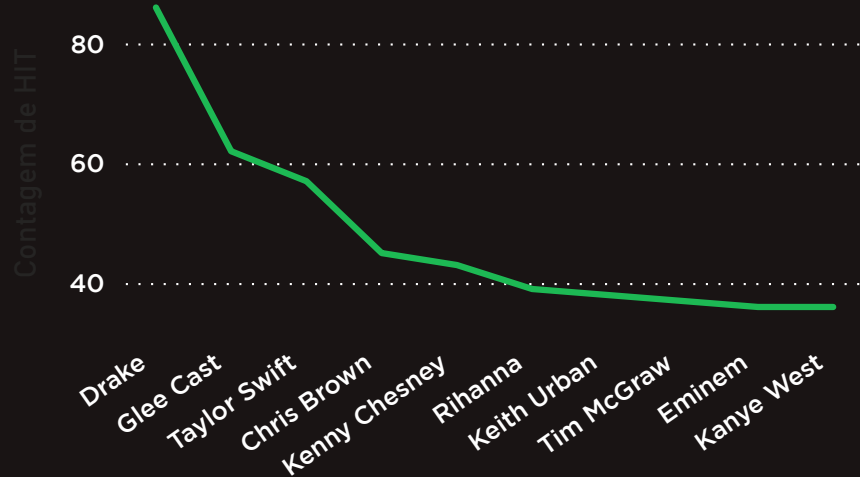
INSTRUMENTALNESS



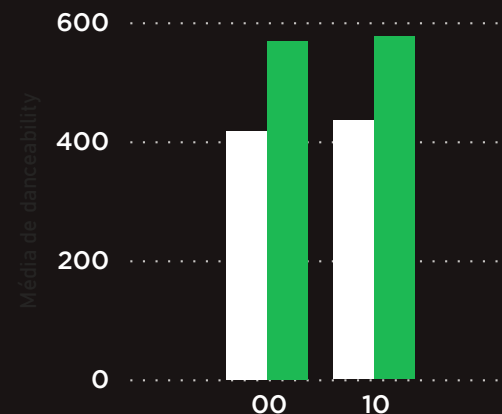
ACOUSTICNESS



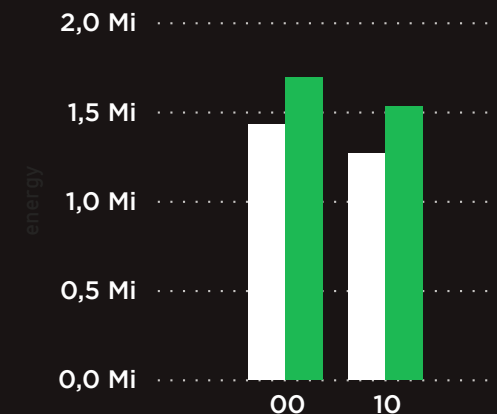
ARTISTAS



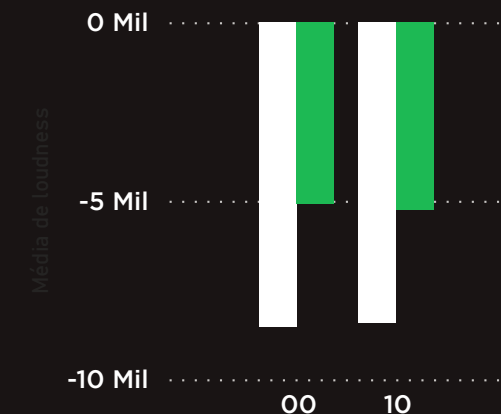
DANCEABILITY



ENERGY



LOUDNESS



Em conclusão, o melhor modelo treinado neste trabalho foi o RandomForest, sendo ligeiramente superior ao XGBoost. Ao treiná-los apenas com as duas décadas mais recentes, ambos tiveram uma melhora no desempenho. Também é possível que essa melhora tenha acontecido devido as bases utilizadas serem menores, pois foram retiradas quatro décadas da análise.

Em relação aos dados, é fato que um alto percentual de nulidade no dataset Last.fm influenciou nas análises, pois mais de oito mil músicas foram excluídas, sendo prejudicial não só ao modelo (ao impossibilitar a análise dessas músicas), como também à análise por gênero e país.

Em termos de experiências futuras, seria interessante investigar a influência da gravadora e a presença na mídia social com relação ao sucesso da música, pois acrescentaria um contexto social ao modelo.

Como suporte para a tomada de decisões importantes, a solução mais credenciada é integrar essas tecnologias com as características humanas, como dos produtores musicais, por exemplo, para obter uma visão mais completa possível sobre a música (NENCINI, 2020).

7. Links

Abaixo estão listados todos os arquivos utilizados no processo de desenvolvimento deste projeto. Os arquivos estão disponíveis em:

[Vídeo no YouTube.](#)

[Repositório GitHub.](#)

Descrição dos arquivos e diretórios da raiz do repositório indicado:

- Dashboard: Diretório contendo a dashboard desenvolvida para este TCC;
- Dataset: Diretório contendo todos os datasets utilizados (originais, tratadas e finais), sendo que o dataset original do Last.fm foi compactado devido ao limite imposto pelo GitHub;
- Fonte: Diretório contendo a fonte utilizada na dashboard e na nuvem de palavras;

- Imagens: Diretório contendo a imagem da nuvem de palavras;
- Jupyter Notebooks: Diretório contendo os notebooks com o desenvolvimento em Python deste trabalho;
- Modelos: Diretório contendo os modelos de machine learning desenvolvidos (RandomForest e XGBoost);

REFERÊNCIAS

NIJKAMP, Rutger. **Prediction of product success: explaining song popularity by audio features from Spotify data**. University of Twente, The Faculty of Behavioural, Management and Social sciences, 2018. Disponível em: <<http://essay.utwente.nl/75422/>>. Acesso em: 26 jun. 2021.

HERREMANS, Dorien; MARTENS, David; SÖRENSEN, Kenneth. **Dance Hit Song Prediction**. Journal of New Music Research: Taylor & Francis, 2014. Disponível em: <<http://dx.doi.org/10.1080/09298215.2014.881888>>. Acesso em: 26 jun. 2021.

AGUIAR, Luis; WALDFOGEL, Joel. **Platforms, Promotion, and Product Discovery: Evidence from Spotify Playlists**. National Bureau of Economic Research, 2018. Disponível em: <<https://www.nber.org/papers/w24713>>. Acesso em: 27 jun. 2021.

AL-BEITAWI, Zayd; SALEHAN, Mohammad; ZHANG, Sonya. **What Makes a Song Trend? Cluster Analysis of Musical Attributes for Spotify Top Trending Songs**. Journal of Marketing Development and Competitiveness, 2020. Disponível em: <<https://search.proquest.com/openview/41a37854d2c51ea29ea8773c768728bb/1?pq-origsite=gscholar&cbl=536309>>. Acesso em: 27 jun. 2021.

NENCINI, Irene. **"Hit o "flop"? Analisi dei fattori determinanti il successo dei brani musicali su Spotify**. Università Degli Studi Di Pisa, Dipartimento Di Economia e Management, 2020. Disponível em: <<https://etd.adm.unipi.it/t/etd-09162020-140650/>>. Acesso em: 27 jun. 2021.

SUN, Hyojung. **Case study—Spotify**. Digital Revolution Tamed: Springer, 2019. Disponível em: <https://doi.org/10.1007/978-3-319-93022-0_5>. Acesso em: 28 jun. 2021.

MIDDLEBROOK, Kai; SHEIK, Kian. **Song hit prediction: Predicting billboard hits using Spotify data**. Cornell University, 2019. Disponível em: <<https://arxiv.org/abs/1908.08609>>. Acesso em: 28 jun. 2021.

PÉREZ-VERDEJO, J. Manuel; PIÑA-GARCÍA C. A.; OJEDA, Mario Miguel; RIVERA-LARA, A.; MÉNDEZ-MORALES, L. **The rhythm of Mexico: an exploratory data analysis of Spotify's top 50**. Journal of Computational Social Science: Springer, 2021. Disponível em: <<https://doi.org/10.1007/s42001-020-00070-z>>. Acesso em: 28 jun. 2021.

BAHULEYAN, Hareesh. **Music genre classification using machine learning techniques**. Cornell University, 2018. Disponível em: <<https://arxiv.org/abs/1804.01149>>. Acesso em: 28 jun. 2021.

INTERIANO, Myra; KAZEMI, Kamyar; WANG, Lijia; YANG, Jienian; YU, Zhaoxia. **Musical trends and predictability of success in contemporary songs in and out of the top charts**. Royal Society Open Science, 2018. Disponível em: <<https://doi.org/10.1098/rsos.171274>>. Acesso em: 28 jun. 2021.

LÓPEZ, Antonio Caparrini; MOLINA, Laura Pérez. **Clasificador de subgéneros de música electrónica**. Universidad Complutense de Madrid, 2017. Disponível em: <<https://eprints.ucm.es/id/eprint/44672/>>. Acesso em: 28 jun. 2021.

OSHIRO, Thais Mayumi; PEREZ, Pedro Santoro; BARANAUSKAS, José Augusto. **How many trees in a random forest?** International Workshop on Machine Learning and Data Mining in Pattern Recognition :Springer, 2012. Disponível em <https://doi.org/10.1007/978-3-642-31537-4_13> Acesso em: 29 jun. 2021.