Algoritmos de Aprendizado de Máquina para Distinguir Gêneros Musicais

Isabella Crosariol
Instituto de Ciência e Tecnologia
Universidade Federal de São Paulo
São José dos Campos, SP, Brasil
crosariol@unifesp.br

***Resumo*— O uso de plataformas de *streaming* de música já é algo comum no dia a dia das pessoas. Neste trabalho, serão analisados os dados de músicas da database do Spotify, juntamente de 3 diferentes algoritmos de aprendizado de máquina que tem como objetivo classificar o gênero de uma ou mais músicas, baseando-se em seus atributos.**

***Abstract*—The use of music streaming platforms is something already common on people's daily lifes. In this essay, music data from Spotify's database will be used and analyzed, along with 3 different machine learning algorithms which have the objective of sorting the genre of one or more songs, using as a base its attributes.**

Keywords—Música, Gêneros Musicais, Atributos Musicais, Algoritmos de Aprendizado de Máquina, Acurácia

I. INTRODUÇÃO

O uso de plataformas de *streaming* de áudio (músicas, podcasts) tem se tornado cada vez mais comum dentro do nosso dia a dia, com a plataforma *Spotify* sendo a mais popular entre usuários do mundo todo – tendo mais que o dobro de assinantes que sua competidora mais próxima, *Apple Music*[1].

O uso de inteligência Artificial para a automatização de reconhecimento de parâmetros não é uma novidade dentro do meio de plataformas de streaming: O Spotify já usava algoritmos para isso em 2016, com suas playlists diárias personalizadas, que buscavam reunir as músicas favoritas de um usuário numa só playlist. Para tarefas como essa, é imprescindível que seja feito um algoritmo que seja capaz de eficientemente classificar músicas – seja por artista, gênero, ou aspectos mais específicos, como batidas por minuto ou quão alegre uma música é.

O objetivo deste trabalho é aplicar diversos algoritmos de aprendizado de máquina aplicados à um dataset de músicas do Spotify, com a finalidade de adquirir um melhor conhecimento sobre o funcionamento de tais algoritmos tal como seu desempenho quando aplicado numa base de dados extensa e, ao menos à primeira vista, com poucos padrões analisáveis entre seus elementos.

Os algoritmos usados foram os seguintes:

- **K-Nearest Neighbors (KNN)**: Usa os vizinhos mais próximos para realizar a classificação por votação.
- **Hunt:** Utiliza uma árvore de decisão de maneira recursiva.
- **Support Vector Machine (SVM):** tem como objetivo encontrar um hiperplano num plano N-dimensional que consiga "separar" os elementos de um conjunto em um ou mais tipos.

II. METODOLOGIA

O principal objetivo deste trabalho foi desenvolver diferentes algoritmos para a classificação de gêneros musicais, tal como a análise dos resultados de cada algoritmo.

A. Dataset

Um dataset consiste num conjunto de dados, agrupado em uma ou mais tabelas, criado a fim de juntar quaisquer dados que forem julgados relevantes na análise de um dado tema.

O dataset utilizado neste trabalho trata-se de um dataset de músicas do Spotify, com mais de 200 mil faixas e 26 gêneros musicais. O dataset possui somente uma tabela, com as seguintes colunas[2]:

- Genre (String): gênero musical atribuído à faixa, num total de 26;
- Artist_name (String): nome do(s) artista(s) ou banda;
- Track_name (String): nome da faixa;
- Track_id (String): id automaticamente atribuído à faixa pelo Spotify;
- Popularity (float): valor entre 0 e 100, com 100 sendo o mais popular, baseado no número de vezes totais em que a faixa foi tocada e quão recente são esses ouvintes;
- Acousticness (float): medida de confiança de 0.0 a 1.0 de se a faixa é acústica ;
- Danceability (float): descreve quão apropriada a música é para se dançar;
- Duration_ms (int): duração da faixa em milissegundos;
- Energy (float): medida de 0.0 a 1.0 que descreve intensidade e atividade (faixas com energy alta são mais rápidas e barulhentas);
- Instrumentalness (float): medida de 0.0 a 1.0, descreve se uma faixa contém vozes ou não. "Oh" e "Ah" são considerados instrumentos;
- Key (String): descreve a nota dentro da escala musical na qual a faixa está. São usadas as 7 primeiras letras do alfabeto, conforme é convenção nos Estados Unidos;
- Liveness (float): medida de 0.0 a 1.0, detecta a presença de uma audiência ao vivo na faixa;

- Loudness (float): medida em geral de -60 a 0, representa o volume geral da faixa em decibéis (dB);
- Mode (String): indica se a música está na escala maior (major) ou menor (minor);
- Speechiness (float): medida de 0.0 a 1.0, detecta a presença de palavras faladas. Faixas exclusivamente faladas (como um podcast) terão um speechiness acima de 0.66;
- Tempo (float): medida em geral de 30 a 250, descreve o tempo de uma faixa em batidas por minuto (BPM);
- Time_signature (String): medida em geral de ¾ a 4/4, representa quantas batidas estão presentes num compasso.
- Valence (float): medida de 0.0 a 1.0, descreve a positividade de uma música.

As colunas referentes ao nome do artista, id da faixa e nome da faixa foram excluídas da tabela dentro do código, pois não foram relevantes para a análise.

Dentro da categoria de gêneros, 2 gêneros foram assimilados à outros similares: Rap (assimilado à Hip-Hop) e Reggaeton (assimilado à Reggae). Além disso, os valores das colunas mode e key, originalmente do tipo strings, foram convertidos em floats, para que todos os valores pudessem ser analisados graficamente em conjunto. "Major" para 1 e "Minor" para 0, e a escala musical foi convertida de C para 1 até B para 12, em ordem crescente. Músicas "duplicadas" (que existiam em mais de um gênero simultaneamente) foram deletadas para não prejudicar a análise de cada gênero, com o critério de escolha para a música a permanecer no dataframe sendo a primeira instância da mesma. Porém, é importante notar que essa escolha causou uma certa distorção na quantidade de músicas de alguns gêneros musicais, visto que não houve um balanceamento na hora de excluir as duplicatas.

Segue a proporção de distribuição de gêneros musicais no dataframe:

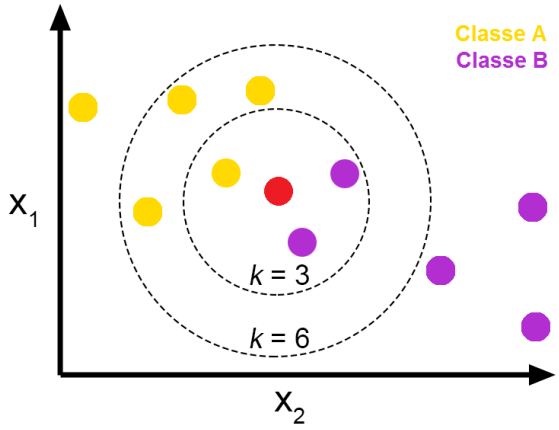
Tabela. 1 - Distribuição de gêneros musicais no dataframe

	Antes	Depois
A Capella	0.06%	0.08%
Pop	5.4%	2.1%
Rock	5.3%	2.3%
Soul	5.2%	3.5%
R&B	5.2%	3.9%
Country	5%	5.4%
Dance	5%	5.9%
Ska	5.1%	5.9%
Folk	5.4%	5.9%
Jazz	5.4%	6%
Opera	4.8%	6.1%
Blues	5.2%	6.3%
Classical	5.3%	6.4%
Anime	5.1%	6.6%
Hip-Hop	10.7%	6.7%
Alternative	5.3%	6.7%
Electronic	5.4%	6.7%
Reggae	10.2%	12.8%

B. Algoritmos

Como já mencionado na introdução, os algoritmos implementados pra este projeto foram o KNN, Hunt e SVM.

Fig. 1 - Algoritmo KNN [3]



O **KNN** usa os k-vizinhos mais próximos para classificar um elemento; note que esse algoritmo assume que um elemento próximo do qual queremos analisar será do mesmo tipo, o que – como veremos mais adiante – nem sempre é verdade. O KNN verifica a proximidade entre os elementos com duas possíveis técnicas:

- Distância Euclidiana:

Entre dois pontos p_1 e p_2,

$d = \sqrt{\sum_{k=1}^{n}(p_1 k - p_2 k)^2}$

- Distância Manhattan:

Entre dois pontos p_1 e p_2,

$d = \sum_{k=1}^{n} |p_1 - p_2|$

O algoritmo de **Hunt** é implementado através de uma árvore de decisão, dividindo o dataset de treino em subsets cada mais mais específicos (ou "puros"). Diferente do KNN, no qual a semelhança é analisada pela combinação de fatores como um só, no Hunt, as características de um elemento devem ser analisadas individualmente, em etapas. O árvore de decisão implementada após análise dos dados obtidos foi a seguinte (árvore de decisão dividida em partes para melhor visualização):

Fig. 2 - Árvore de decisão (1/5)

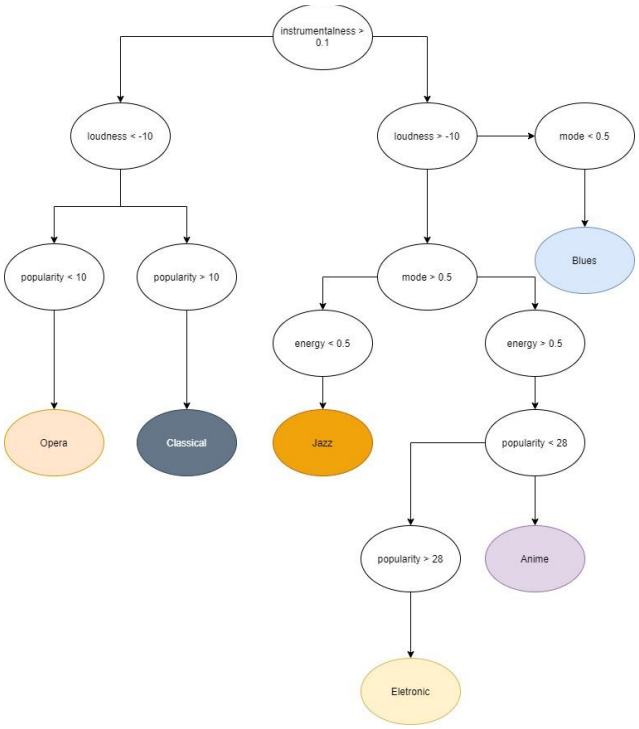


Fig. 3 - Árvore de decisão (2/5)

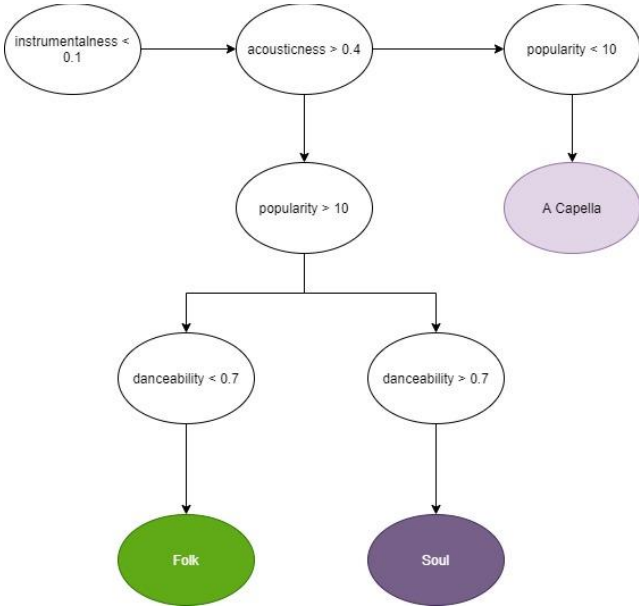


Fig. 4 - Árvore de decisão (3/5)

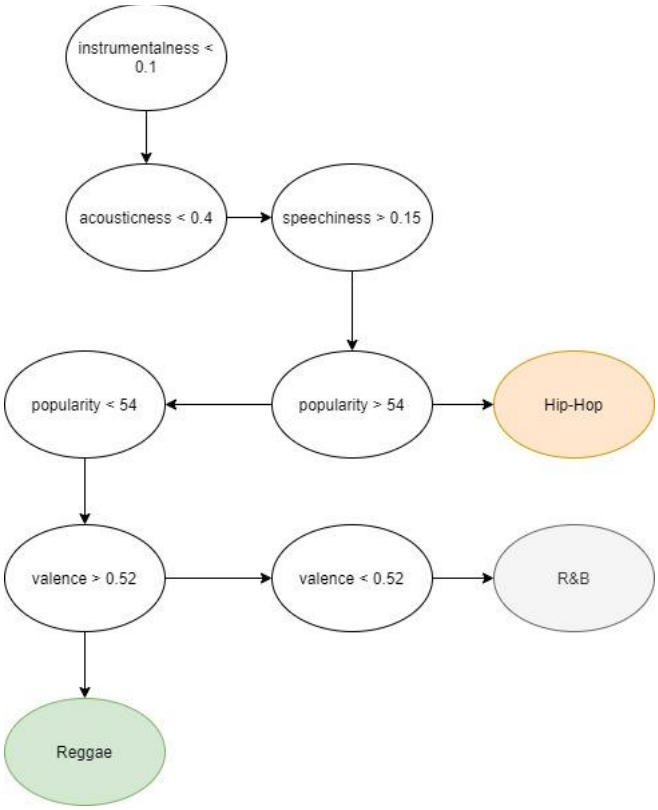


Fig. 5 - Árvore de decisão (4/5)

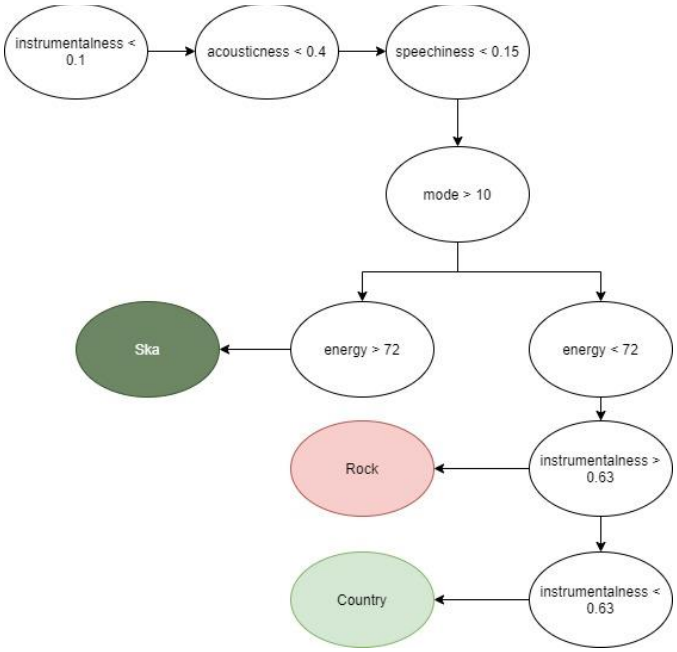
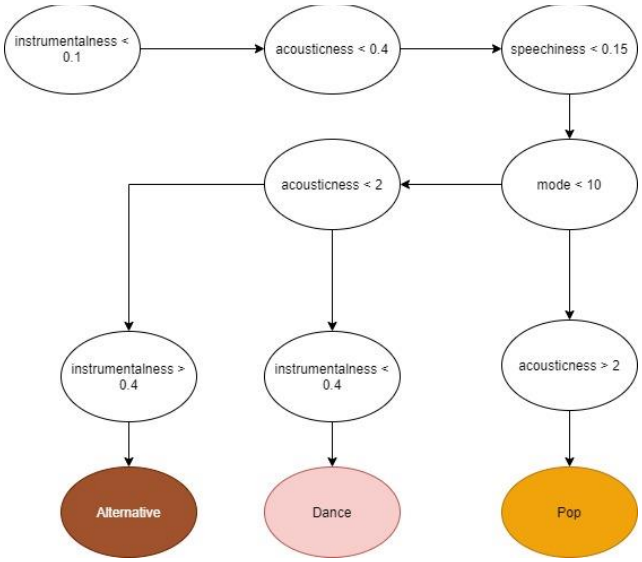
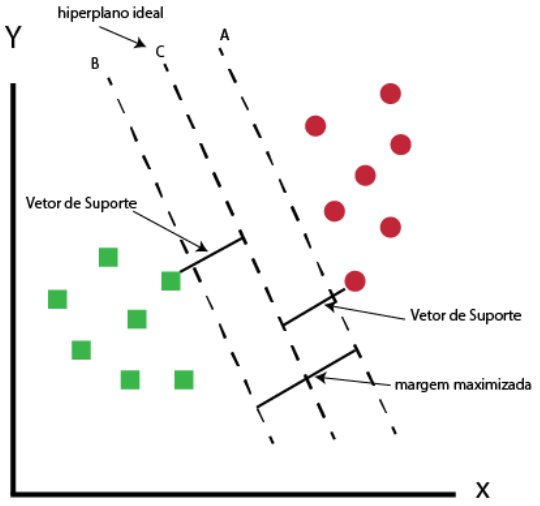


Fig. 6 - Árvore de decisão (5/5)



O **SVM**, ou máquina de vetores de suporte, cria um hiperplano (uma fronteira) num plano N-Dimensional entre os elementos a serem analisados. O melhor hiperplano possível será aquele que tem a maior distância (também chamada de margem) entre as classes de elementos.

Fig. 7 – Algoritmo SVM [4]



A implementação do SVM foi feita através da biblioteca sklearn em Python.

C. *Bibliotecas Utilizadas*

Durante o desenvolvimento deste trabalho, foram utilizadas as seguintes bibliotecas:

- **NumPy**: permite trabalhar com arrays e matrizes multidimensionais, oferecendo também diversas funções matemáticas capazes de trabalhar com arrays e matrizes.
- **Pandas**: permite a manipulação e análise de dados, permitindo manipular tabelas.
- **Matplotlib**: traz funções de plotagem 2D com uma grande variedade de opções de gráficos.
- **Scikit-Learn**: é uma biblioteca de aprendizado de máquina para Python, que traz diversos algoritmos de classificação, regressão, agrupamento, entre outros.

III. Análise Experimental

A. *Conjunto de dados*

O Dataset utilizado**, Spotify Tracks DB**, contém 232.725 faixas, tendo sido separado aproximadamente 10.000 músicas para cada gênero musical (que no total são 26).

Foram feitos gráficos de todos os atributos para os gêneros a serem analisados. Além do gráfico de barras, para os atributos que visivelmente apresentavam valores com grande variação também foram feitos gráficos de diagrama de caixa.

Fig. 8 – Popularidade

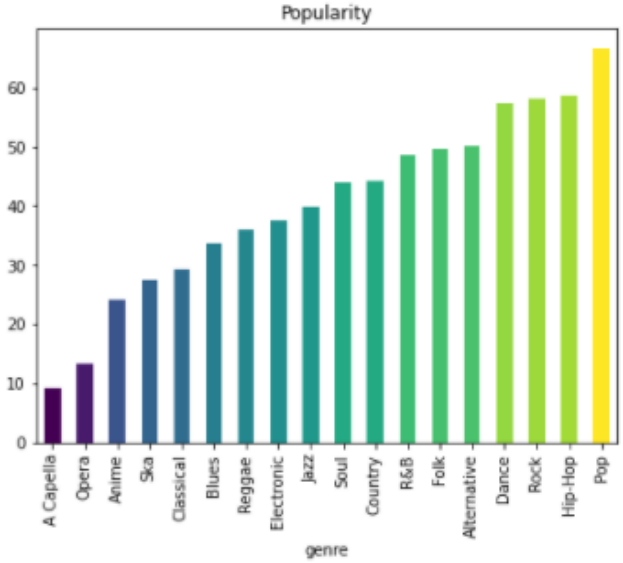
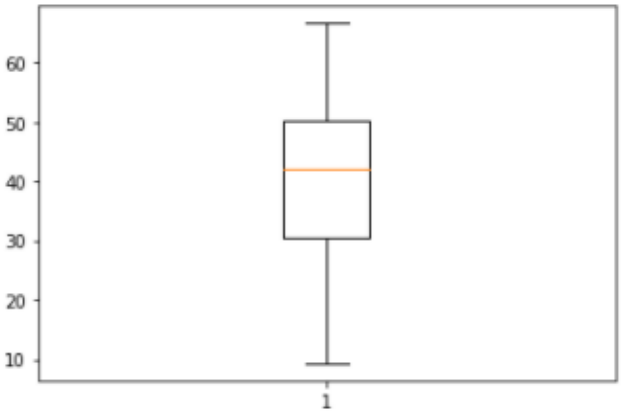


Fig. 9 – Diagrama de caixa popularidade



Na categoria popularidade, nota-se uma clara distinção entre alguns gêneros, como A Capella e Opera (baixa popularidade) e Pop e Hip-Hop (alta popularidade).

Identify applicable funding agency here. If none, delete this text box.

Fig. 10 – Nível de acústico

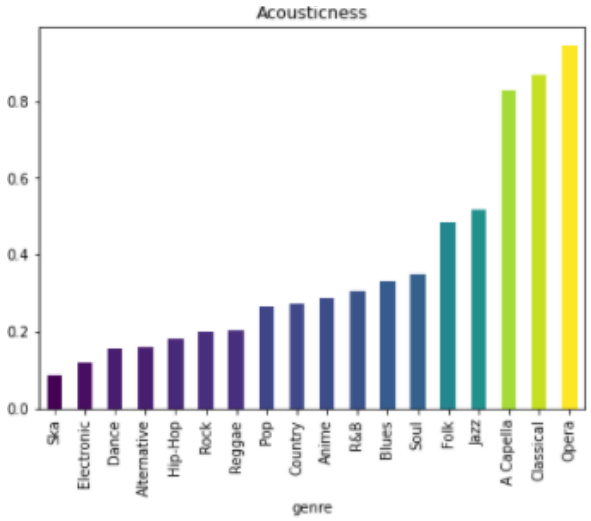
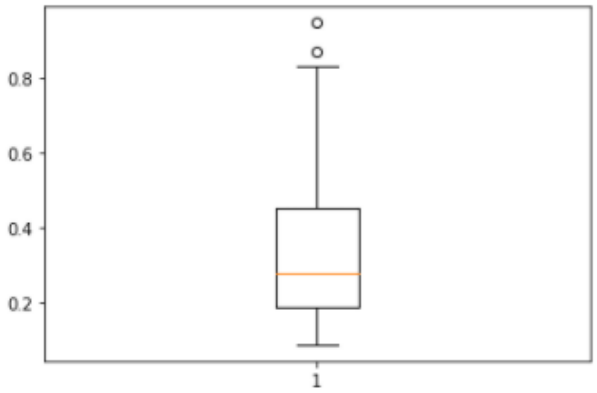


Fig. 11 – Diagrama de caixa nível de acústico



Os valores de níveis de acústica são previsíveis; valores altos para, principalmente, Opera, Classical e A Capella, e baixo para gêneros como Ska e Electronic.

Fig. 12 – Nível de "dançabilidade"

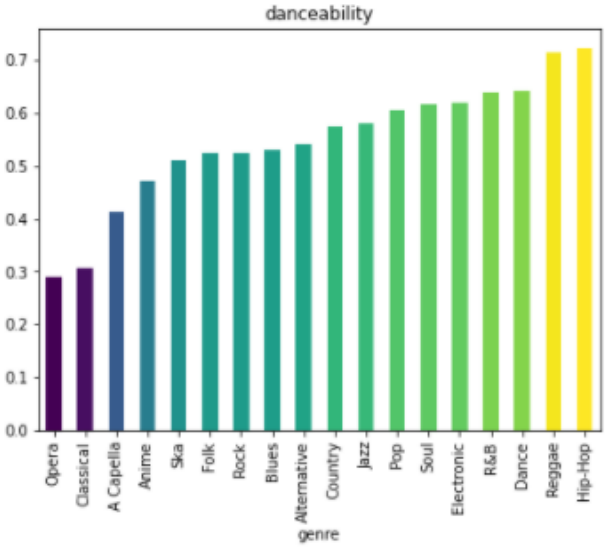
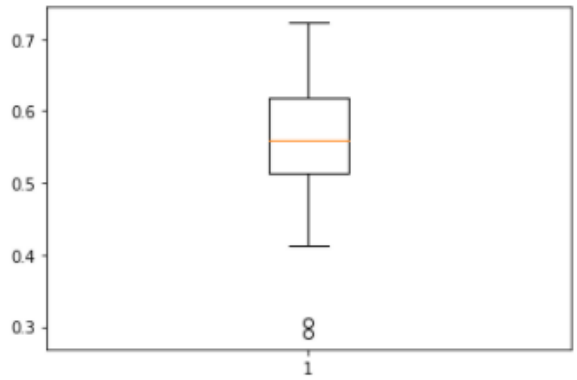
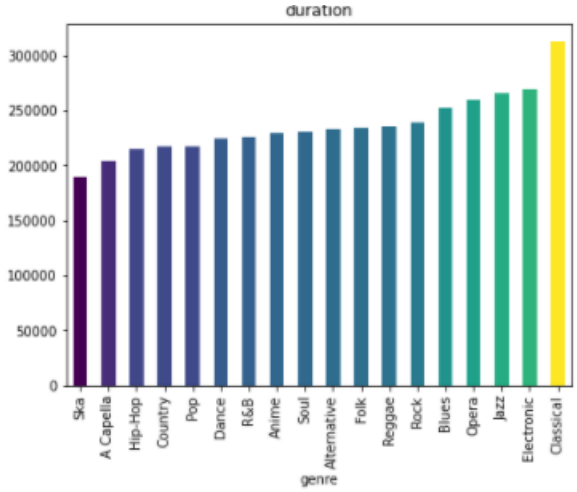


Fig. 13 – Diagrama de caixa nível de "dançabilidade"



Também conforme o esperado, Opera, Classical e A Capella tem valores baixo para esse atributo. Os outros gêneros, em geral, obtiveram valores altos (acima de 0.5).

Fig. 14 – Duração (em milissegundos)



Quanto à duração, o gênero musical que irá destoar dos outros será o Classical, que conta com diversas faixas com duração maior que 5 minutos (300.000 milissegundos).

Fig. 15 – Energia

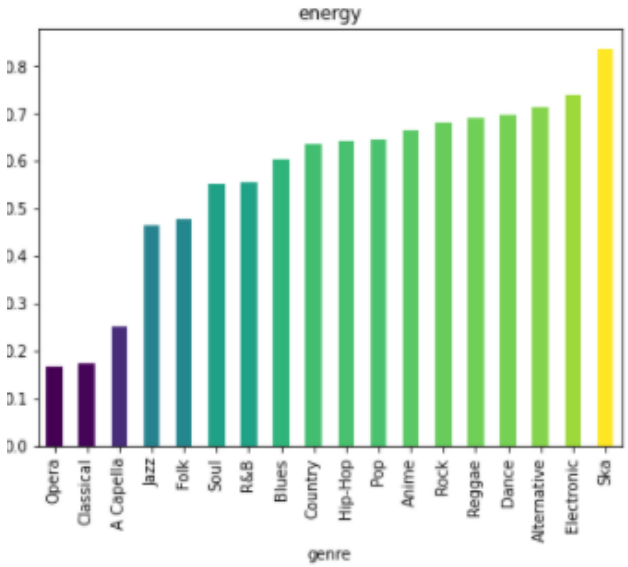
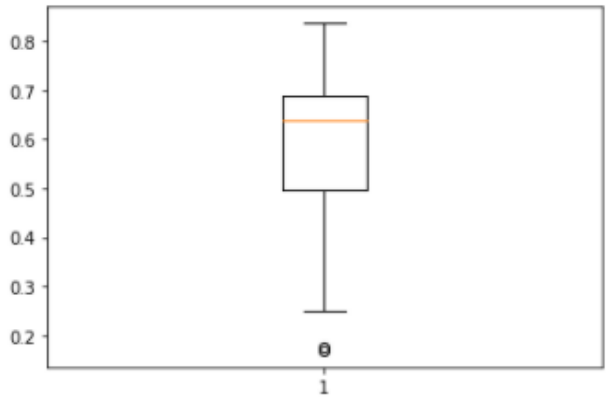


Fig. 16 – Diagrama de caixas energia



Novamente, Opera, Classical e A Capella destoam dos demais gêneros musicais, com os três tendo baixos níveis de energia. Os demais gêneros musicais apresentam, em geral, níveis semelhantes.

Fig. 17 – Nível de instrumental

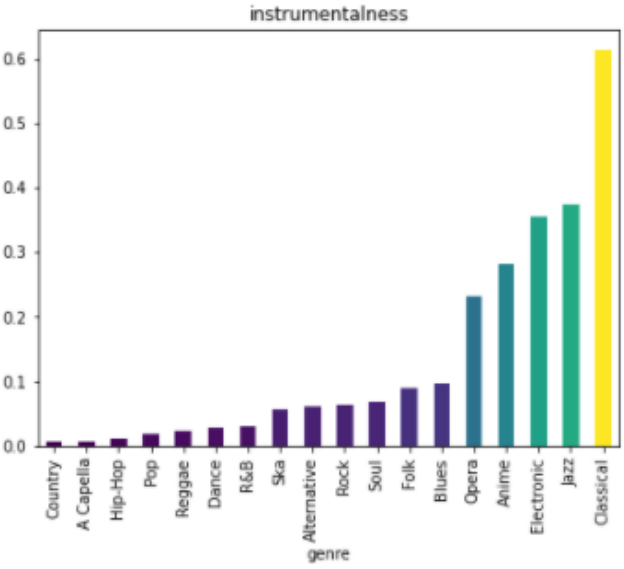
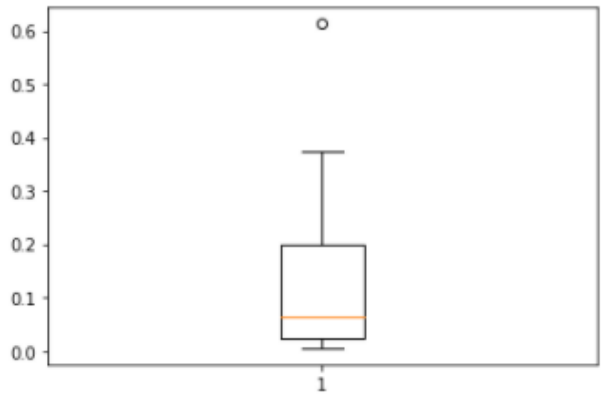
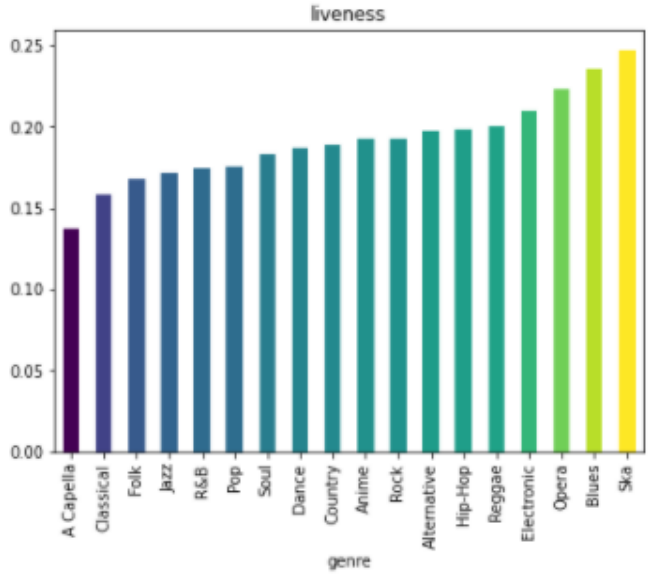


Fig. 18 – Diagrama de caixa nível de instrumental



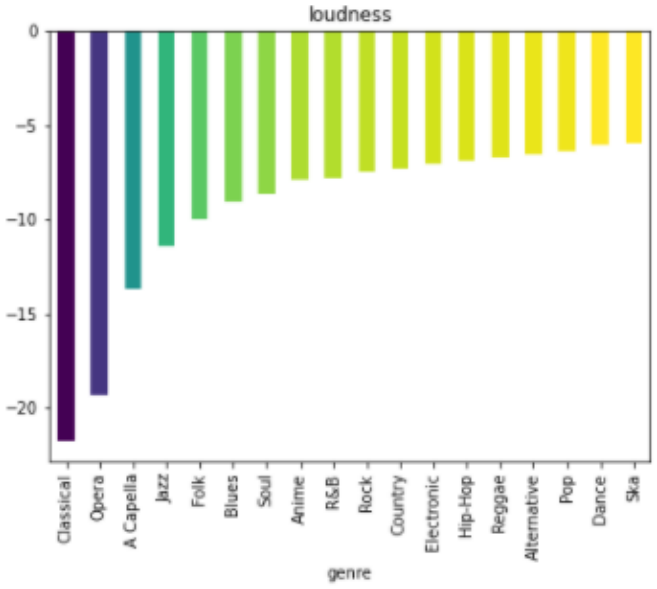
A maioria dos gêneros musicais apresentam valores consideravelmente baixos (menores que 0.1) de nível de instrumental, com as exceções mais notáveis sendo Clasical, Jazz e Electronic. Diferente dos demais gráficos em que o grupo de gêneros destoantes dos demais eram Classical, A Capella e Opera, desta vez temos uma amostra mais diferenciada.

Fig. 19 – Nível de “ao vivo”



Em geral, esse atributo não apresenta grandes variações entre os gêneros musicais.

Fig. 20 – Nível do volume



Mais uma vez, os gêneros que se diferenciam dos ademais são A Capella, Classical e Opera. Com exceção desses 3, os demais gêneros possuem, em geral, valores médio-baixos.

Fig. 21 – Nível de “fala”

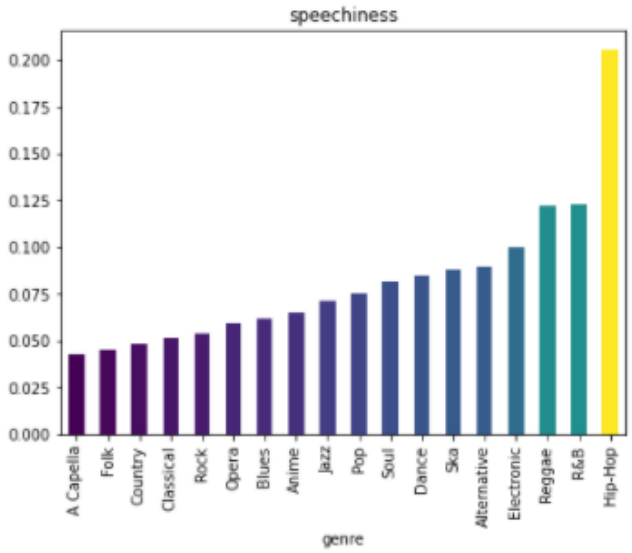
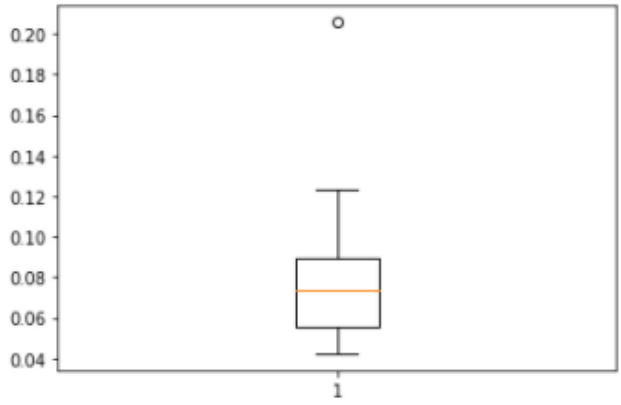
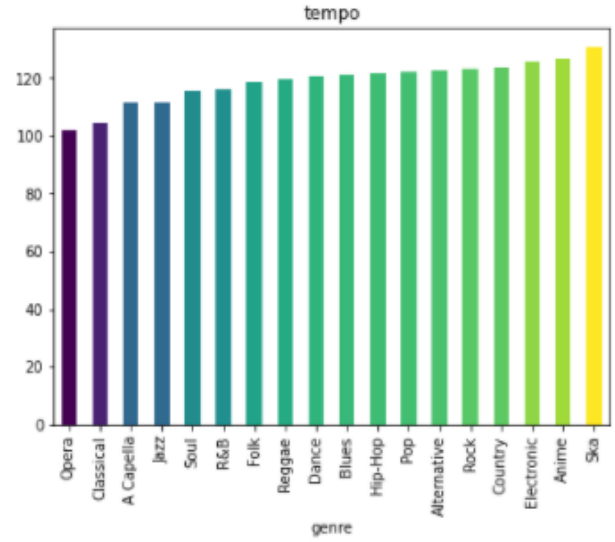


Fig. 22 – Diagrama de caixa nível de “fala”



Com exceção de Hip-Hop, a maior parte dos gêneros apresenta valores semelhantes.

Fig. 23 – Tempo



Os valores entre cada gênero deste atributo são os que menos variam entre si.

Fig. 24 – Nível de positividade

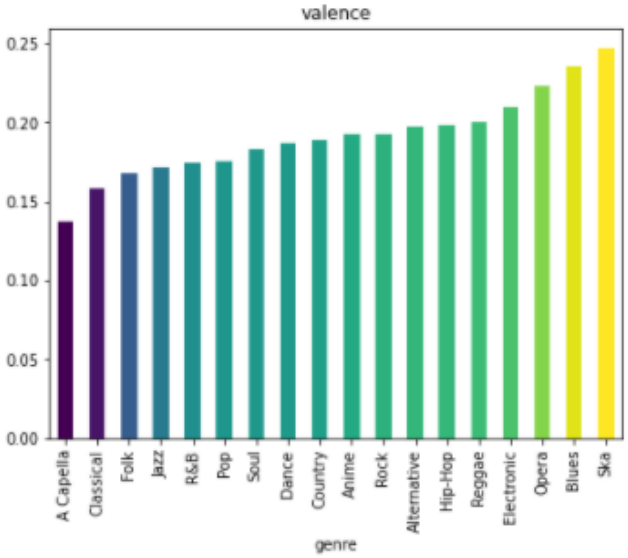
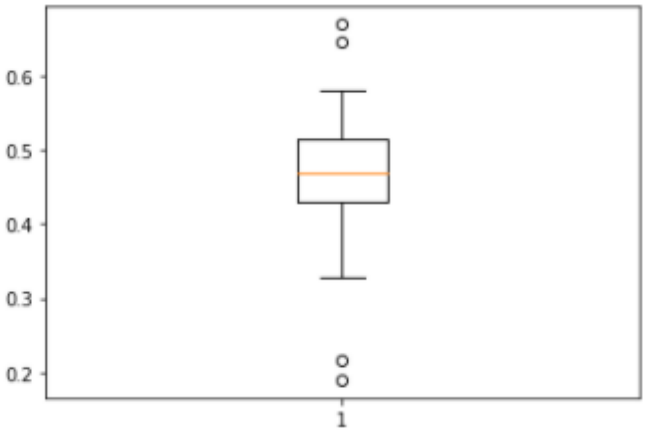


Fig. 25 – Diagrama de caixa nível de positividade



Não há uma grande variação entre todos os gêneros em geral para este atributo (com a maior parte dos gêneros musicais inclusive tendo valores bem próximos, como pode ser observado no diagrama de caixa), mas pode-se notar uma diferença notável entre os gêneros de maior positividade (Ska, Blues) e menor positividade (A Capella, Classical).

Fig. 26 – Escala (maior/menor)

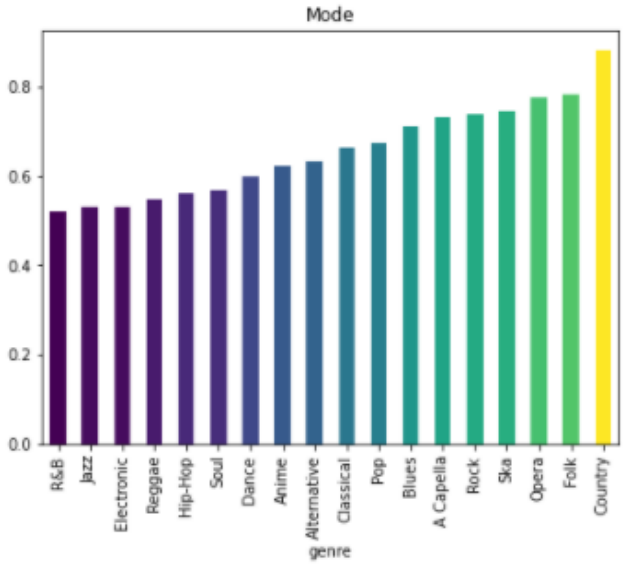
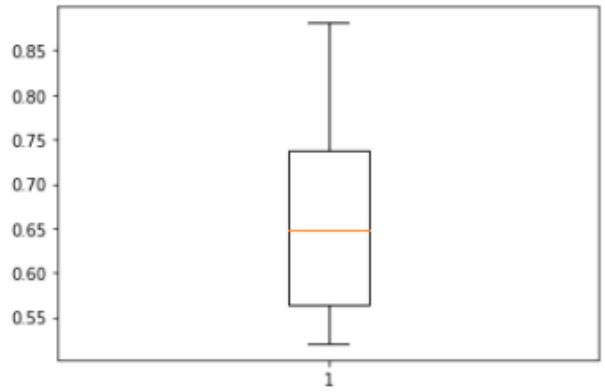


Fig. 27 – Diagrama de caixa escala (maior/menor)



Neste caso ocorre o mesmo que no gráfico anterior; em geral, não há notável variação, mas se pode notar uma diferença entre os gêneros com maior valor (Country, Folk) e menor valor (R&B, Jazz).

B. Configuração do algoritmo e do ambiente computacional

O algoritmo foi construído totalmente dentro da plataforma Kaggle, que possui as seguintes características [5]:

- Cpu: modelo Intel® Xeon®, 4 processadores de frequência 2.30GHz;
- Espaço de Disco: 5GB
- Memória: 16GB (dos quais aproximadamente 410MB são usados)

Demais informações sobre o kernel do Kaggle não foram encontradas.

C. Critérios de Análise

Em cada execução, são escolhidas linhas aleatórias da tabela, através da função sample.

Tabela 2 – Testes de acurácia (valor de teste constante)

Tamanho		Acurácia			
		KNN		Hunt	SVM
		Euclidiano	Manhattan		
Treino	47	12.5%	15%	67.5%	1%
Teste	40				
Treino	53	7.5%	10%	62.5%	6%
Teste	40				
Treino	94	17.5%	22.5%	65%	20%
Teste	40				
Treino	269	10%	10%	72.5%	13%
Teste	40				
Treino	376	5%	12.5%	55%	20%
Teste	40				
Treino	673	17.5%	20%	62.5%	10%
Teste	40				
Treino	1077	25%	27.5%	47.5%	16%
Teste	40				
Treino	1346	17%	23%	64%	14%
Teste	40				
Média		14	17.56	62.06	12.5
Mediana		14.75	17.5	63.25	13.5
Desvio Padrão		6.5	6.6	7.68	6.63

Tabela 3 – Testes de acurácia (valor de treino constante)

Tamanho		Acurácia			
		KNN		Hunt	SVM
		Euclidiano	Manhattan		
Treino	1211	0%	10%	60%	33%
Teste	10				
Treino	1211	24%	24%	62%	12%
Teste	37				
Treino	1211	26%	10%	69%	6%
Teste	52				
Treino	1211	20%	26%	67%	19%
Teste	67				
Treino	1211	16%	21%	71%	10%
Teste	121				
Treino	1211	22%	25%	65%	19%
Teste	397				
Treino	1211	21%	22 %	64%	11%
Teste	669				
Treino	1211	23%	24%	65%	15%
Teste	1063				
Média		19	20.25	65.14	15.62
Mediana		24.5	23	65	13.5
Desvio Padrão		8.22	6.51	3.80	8.3

Tabela 4 – Testes de acurácia (valores constantes, sementes diferentes)

Tamanho		Acurácia			
		KNN		Hunt	SVM
		Euclidiano	Manhattan		
Treino	161	32%	33%	64%	14%
Teste	59				
Treino	161	23%	23%	60%	22%
Teste	59				
Treino	161	21%	25%	61%	8%
Teste	59				
Treino	161	18%	25%	68%	10%
Teste	59				
Treino	161	25%	26%	60 %	4%
Teste	59				
Treino	161	15%	13%	70%	14%
Teste	59				
Treino	161	21%	20%	66%	10%
Teste	59				
Treino	161	16%	15%	66%	18%
Teste	59				
Média		21.375	22,5	64.375	12.5
Mediana		21	24	65	12
Desvio Padrão		5.47	6.41	3.77	5.73

D. Resultados e Discussão

Fig. 28 – Gráfico de resultados (tabela 2)

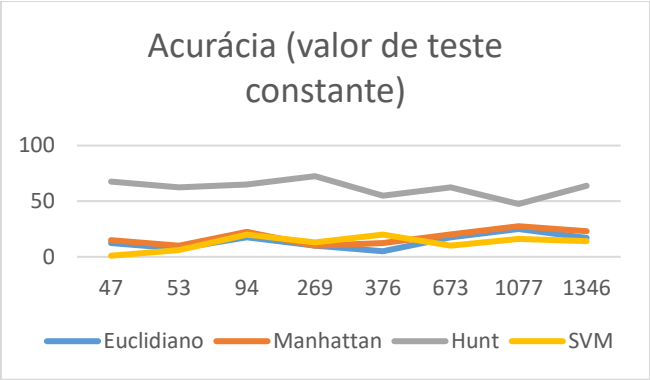


Fig. 29 – Gráfico de resultados (tabela 3)

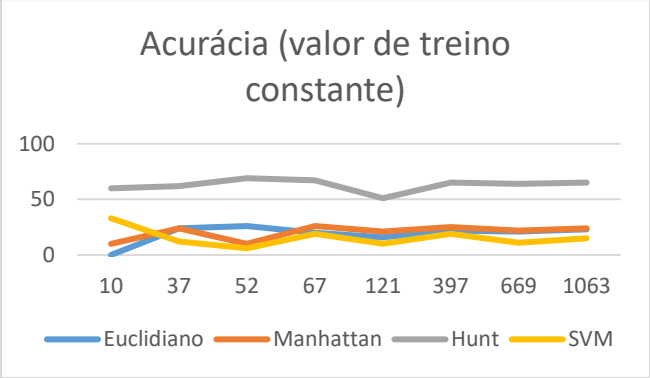
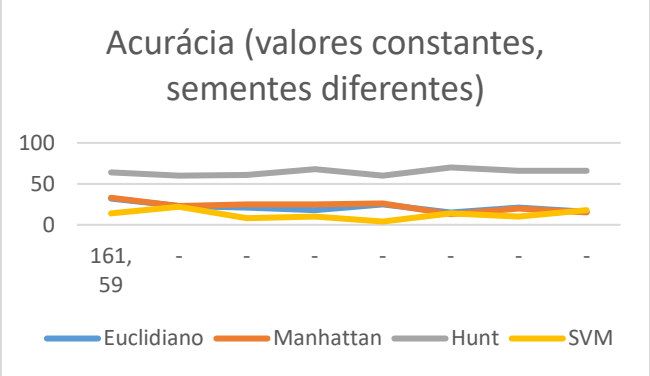


Fig. 30 – Gráfico de resultados (tabela 4)



Podemos observar que, para todos os casos, o algoritmo de Hunt se demonstrou como mais eficiente por uma margem notável.

O mais provável fator para isso é o fato de os dados do dataframe não seguirem um "padrão", por assim dizer: apesar de certos gêneros musicais terem caraterísticas marcantes, não há, de fato, nada que se aproxime de u conjunto de regras rígidas de definição para qualquer gênero musical, o que acaba permitindo inúmeras variações dentro de um mesmo gênero, que por fim permite que duas músicas de gêneros musicais distintos tenham características extremamente semelhantes – daí tornado o KNN e SVM ineficazes, o KNN pois um elemento estar próximo daquele que está sendo analisado raramente vai significar que de fato é da mesma classe num ambiente multi-dimensional[6], e SVM pois com tantos elementos de classes diferentes em posições "aleatórias", é difícil criar uma fronteira que demarque a divisão entre duas (ou mais) classe(s) de forma correta[7]. Como o algoritmo de Hunt é feito através da análise em conjunto de todos os parâmetros de cada gênero, faz sentido ele ser o que possui maior acurácia entre os algoritmos analisados.

IV. Conclusões

A música é algo que já faz parte da rotina do brasileiro: segundo uma pesquisa feita pela Opinion Box, 80% das pessoas ouvem música todo dia, e desse grupo, 32% diz ouvir música o dia todo. Seja durante os estudos, trabalho ou enquanto se limpa a casa, é um hábito comum não somente dedicar um tempo para ouvir música, mas também ouvir enquanto se realiza alguma outra tarefa[8].

Considerando principalmente a época na qual estamos, em que shows e eventos musicais já não ocorrem presencialmente faz mais de um ano, serviços de streaming se tornam cada vez mais necessários para qualquer pessoa que queira ouvir música[9]; daí, surge a necessidade cada vez mais de não só aprimorar mais quaisquer algoritmos já existentes como também buscar tentar criar-se novos algoritmos, almejando sempre resultados mais efetivos e execução rápida.

Através do estudo realizado, foi possível observar que devido à alta complexidade dentro dos parâmetros de música, desenvolver algoritmos eficazes que classifiquem múltiplos gêneros é um grande desafio; sem haver de fato um grande padrão a ser analisado dentro de cada gênero musical, a melhor opção é fazer uma análise "personalizada", levando em conta grupos que possuam características em comum.

Referências

[1] WALLACH, Omri. Which streaming service has te most subscriptions? World Economic Forum, 2021. Disponível em: <Which streaming service has the largest subscriber base? | World Economic Forum (weforum.org)>. Acesso em 26 de julho de 2021

[2] SPOTIFY c2021, Web API Reference, disponível em: < https://developer.spotify.com/documentation/web-api/reference/#endpoint-get-audio-features>. Acesso em 24 de julho de 2021

[3] JOSÉ, Ítalo. KNN (K-Nearest Neighbors) #1. Medium, 2018. Disponível em: <https://medium.com/brasil-ai/knn-k-nearest-neighbors-1-e140c82e9c4e>. Acesso em 25 de julho de 2021.

[4] CAVALCANTI, Toti. Aula 08 – Scikit-Learn – máquina de vetores de suporte, s.d., disponível em: <https://www.codigofluente.com.br/aula-08-scikit-learn-maquina-de-vetores-de-suporte/>. Acesso em 25 de julho de 2021.

[5] KAZEMNEJAD, Amirhossein.How to do Deep Learning research with absolutely no GPUs – Part 2. Amirhossein Kazemnejad's Blog, 2019. Disponível em: < How to do Deep Learning research with absolutely no GPUs - Part 2 - Amirhossein Kazemnejad's Blog>. Acesso em 31 de julho de 2021

[6] Lecture 2: k-nearest neighbors, disponível em: <Lecture 2: k-nearest neighbors / Curse of Dimensionality (cornell.edu)>. Acesso em 30 de julho de 2021.

[7] CHAKURE, Afroz. Support Vector Machines (SVMs). Medium, 2019. Disponível em: <Support Vector Machines (SVMs). Introduction | by Afroz Chakure | DataDrivenInvestor>. Acesso em 30 de julho de 2021.

[8] Consumo de Música no Brasil. Abramus, 2021. Disponível em: <https://www.abramus.org.br/noticias/16444/consumo-de-musica-no-brasil/>. Acesso em 31 de julho de 2021.

[9] TSENG, Andrew. 2020 Wrapped: Is Spotify Stock a Buy? The Motley Fool, 2020. Disponível em: <2020 Wrapped: Is Spotify Stock a Buy? | The Motley Fool>. Acesso em 1 de agosto de 2021.