Centro Universitário de Brasília
Instituto CEUB de Pesquisa e Desenvolvimento - ICPD

SPOTIFY SEQUENTIAL SKIP PREDICTION CHALLENGE: Exploração e construção de modelos de predição do comportamento de saltos de faixas musicais.

Gabriel Rodrigues Alves[1]

RESUMO

O Spotify é uma das maiores plataformas de streaming de música atualmente. Com o crescente número de usuários e artistas, um alto volume de dados são gerados, o que torna a sua exploração imprescindível. Os sistemas de recomendação desempenham um papel importante nestes serviços, principalmente por proporcionar *'Playlists*' personalizadas para seus usuários. Explorar e entender as interações dos clientes com as sessões pode ser benéfico para compreender suas preferências no contexto de cada sessão. Dessa forma, a empresa criou o *Spotify Senquential Skip Prediction Challenge*, um desafio aberto e focado em prever se uma faixa de música em uma sessão será ignorada pelo usuário. O objetivo deste trabalho foi explorar os dados fornecidos e construir um modelo que pudesse prever se um usuário da plataforma iria pular uma faixa, visando um melhor entendimento sobre suas interações com a plataforma em si. Através de exploração e mineração da base de dados para conseguir as melhores características, um pipeline com oito algoritmos de machine learning foi construído para prever os pulos de cada usuário. Desta forma foi possível observar a capacidade que o algoritmo XGBoost tem e sua eficiência para resolver problemas de classificação de dados, apresentando uma boa acurácia para resolução do desafio.

Palavras-chave: Machine learning; pulos de faixa; música; modelos de predição; sistema de recomendação de música.

---

[1] Trabalho apresentado ao Centro Universitário de Brasília (CEUB/ICPD) como pré-requisito para obtenção de Certificado de Conclusão de Curso de Pós-graduação *Lato Sensu* em ... , sob orientação do Prof. André Juan Costa Vieira. ... Banca realizada em 13/09/23, composta pelos professores-avaliadores Prof. Me. Arthur Porfírio e Prof. Dr. Gilson Ciarallo

1 INTRODUÇÃO

O mercado de músicas teve mudanças significativas com a chegada da era digital, o que trouxe o surgimento de uma das plataformas mais influentes nesse ramo: Spotify. De acordo com o site newsroom (2022), a empresa se transformou no maior aplicativo de streaming de áudio do mundo, com 422 milhões de usuários, incluindo 182 milhões de assinantes e mais de 82 milhões de trilhas sonoras em seu domínio.

O Spotify se tornou um dos provedores de música preferidos do público, e com isso seu grande número de dados passou a ser algo de significativa importância para análise e tomada de decisão para sua gestão. Com o amplo número de pessoas procurando e escutando músicas diariamente a plataforma observou uma importância em recomendar as músicas certas para seus usuários. E os métodos de Ciência de Dados Machine Learning passaram a ter um alto valor para esse trabalho (Hurtado; Wagner; Mundada, 2019).

Hurtado, Wagner e Mundada (2019) também apontam o fato de que não há muitas pesquisas em relação a como os usuários interagem com músicas tocadas em sequências ao longo de uma sessão de audição, com isso o comportamento de pular trilhas serve como um poderoso sinal do que o usuário gosta ou não das faixas. Dessa forma os métodos de Machine Learning e Inteligência Artificial passam a ter um grande valor para ajudar a prever esses comportamentos.

Com isso o Spotify criou um desafio de previsão de salto sequencial, que consiste em construir um sistema de predição de trilhas que serão puladas pelos usuários em uma sessão de escuta. A tarefa é prever se faixas serão ignoradas por determinado usuário. Para isso, a empresa disponibilizou informações completas sobre a primeira metade das sessões de escuta de um usuário e a previsão deve ser realizada na segunda metade.

Com sistemas de Machine Learning de classificação é possível entender o comportamento dos usuários, além de criar uma melhor assertividade da plataforma para recomendar faixas de música que realmente possam agradar um determinado usuário através das predições das faixas que ele irá pular.

1.1 JUSTIFICATIVA

Pela ótica social este estudo justifica-se por apresentar em seus resultados o comportamento de usuários em plataformas provedoras de música, ajudando o entendimento também das funcionalidades e ajudando na tomada de decisão delas. Além disso, o projeto pode ajudar em uma melhor tomada de decisão para melhor qualidade de serviços de seus usuários. Com o entendimento de como os usuários respondem a determinadas músicas, os serviços de streaming podem de forma eficiente distribuir músicas certas para determinados usuários de forma assertiva.

Do ponto de vista acadêmico esse trabalho contribuirá para pesquisas acerca da área de Ciência de Dados e Machine Learning, principalmente na área de áudio e música. Visto que existe pouco entendimento em relação a como usuários interagem com faixas de áudio em aplicativos de música, o atual estudo também ajudará no entendimento de desenvolvimento de sistemas de predição deste comportamento.

No aspecto pessoal o estudo busca aprimorar e obter excelência nas práticas de Ciência de Dados e Aprendizado de Máquina tanto no âmbito acadêmico quanto profissional. Além disso, visa explorar a aplicação dessas habilidades na área da música, pela qual o autor do trabalho nutre admiração. O curso oferece a oportunidade de aplicar as ferramentas e conhecimentos adquiridos, enriquecendo a experiência de estudo nesse campo específico.

2 Objetivo geral

O objetivo geral deste trabalho é explorar os dados fornecidos pelo Spotify e criar modelos de *machine learning* que tenham capacidade de prever quais faixas de músicas serão puladas pelos usuários do aplicativo.

2.1 Objetivos específicos

- Analisar o DataFrame oferecido pelo Spotify;
- Identificar os dados importantes a serem usados;

- Minerar e limpar os dados necessários;
- Desenvolver algoritmos de machine learning;
- Treinar os algoritmos nos dados preparados;
- Interpretar os resultados.

Para alcançar esses objetivos, procedeu-se da seguinte maneira. As bases de dados foram obtidas diretamente da plataforma de pesquisa do Spotify, desta forma foi possível visualizar e explorar todas as características dos dados oferecidos. Constatou-se 2 DataFrames diferentes, um contendo características e comportamento dos usuários e outro contendo características das faixas musicais.

Através da exploração dos dados foi encontrado a característica que indicam se os usuários pularam determinada faixa de áudio ou não, assim separada como o vetor de saída e classe alvo para o treinamento dos algoritmos.

Para o restante da base de dados, foi usado métodos de Label Encoder para atribuição de valores numéricos para os dados categóricos, dessa forma facilitando compreensão e o processamento dos dados pelos algoritmos, também foi usado métodos estatísticos e algoritmos de Random Forest e valores de SHAP para identificar as características mais importantes das bases de dados.

Com toda base de dado preparada, um pipeline com oito algoritmos foi construído, contendo os modelos: Regressão logística, Decision Tree, Random Forest, Gradient Boosting, Support Vector Machine (SVM), Naive Bayes, XGBoost e uma Rede Neural Convolucional (CNN) para o treinamento dos dados, dessa forma sendo possível prosseguir com as análises de resultados obtidos.

3 O SPOTIFY

O Spotify é um serviço de streaming de música que revolucionou a forma como as pessoas acessam e consomem música. Fundado em 2006 por Daniel Ek e Martin Lorentzon na Suécia. Foi lançado em 2008, e seus serviços permitiram aos usuários acesso e transmissão de música de forma legal e gratuita, com ajuda de financiamentos e anúncios. Em pouco tempo se tornou uma força dominante no mercado de streaming, proporcionando aos usuários um vasto catálogo com diversidade

de músicas de diferentes artistas e gêneros. O site Canaltech ([2000?]) Afirma também que o software teve um protagonismo, pois a indústria de música estava sofrendo com pirataria desenfreada.

Segundo Mamede (2023) com o surgimento do Napster, milhões de pessoas tiveram a possibilidade de baixar música gratuitamente, para as grandes gravadoras isso passou a ser uma ameaça para seus números de receita e queriam que isso acabasse. De acordo com próprio Daniel Ek na época: “Você nunca pode legislar contra a pirataria. As leis definitivamente podem ajudar, mas isso não resolve o problema. A única maneira de resolver o problema era criar um serviço melhor que a pirataria e, simultaneamente, compensar a indústria musical” (Mamede, 2023).

De acordo com Rettberg e Kolås (2019), o aplicativo desempenhou um papel importante na transição da indústria musical para um modelo baseado em streaming: “O Spotify permitiu que as pessoas ouvissem música por demanda, o que resultou em uma mudança na forma como a música é consumida”.

Após seu lançamento em 2008, começou a ganhar popularidade rapidamente, especialmente na Europa. Em 2011, o serviço foi expandido para os Estados Unidos, que levou a um aumento significativo na base de usuários. Desde então vem crescendo e se tornou um dos principais serviços de streaming de música em todo o mundo (Mamede, 2023).

Uma de suas características é o seu modelo “*freemium*”, que entrega o serviço de forma gratuita com anúncios que são produzidos depois de uma quantidade de músicas ouvidas pelo usuário e uma versão *premium* paga, que oferece recursos adicionais, como o direito de ouvir músicas *offline*, evitar anúncios e acesso a melhores qualidades de áudio. Poyar (2020) aponta que essa abordagem ajudou a atrair uma ampla base de usuários e a converter muitos deles em assinantes *premium*. Para Katz (2010), o êxito do aplicativo se deve pela capacidade de adaptabilidade às mudanças no comportamento dos consumidores. O Spotify entendeu que as pessoas queriam um acesso fácil e rápido à música, e construíram uma plataforma que atende essa demanda, permitindo que os usuários encontrem e ouçam suas músicas favoritas em qualquer lugar e a qualquer momento.

Também se destacou por criar uma rede de característica social e de recursos de compartilhamento. Os usuários podem criar e compartilhar *playlist* com amigos, seguir artistas e receber recomendações baseados em suas atividades e gostos musicais. Além disso, criou alguns recursos como o “*Discover Weekly*", uma *playlist* personalizada de músicas novas para o usuário descobrir, e o "*Release Radar*", que destaca as novas músicas dos artistas que os usuários seguem.

Ao longo dos anos, a plataforma expandiu seu alcance além da música, incluindo podcasts. Em 2019, o Spotify adquiriu as plataformas de produção e distribuição de podcasts Anchor e Gimlet Media. Essas aquisições demonstraram o compromisso em se tornar uma plataforma abrangente e generalizada de áudio.

Jenkins (2014), afirma que o Spotify foi pioneiro em uma nova forma de consumo de música, baseada em conveniência e acessibilidade, foi uma das primeiras plataformas que oferece acesso instantâneo e sob demanda a um vasto catálogo de músicas, permitindo aos usuários criar suas próprias *playlists* e descobrir novos artistas com facilidade.

Com a ascensão do streaming a plataforma é um exemplo de como a indústria da música passou por uma transformação significativa. A empresa enfrentou desafios, como a necessidade de negociações de licenciamento com gravadoras, artistas e concorrência de outros serviços de streaming, mas continuou a inovar e expandir seu serviço pelo mundo.

3.1 Importância do “pulo” de faixas

Ao compreender a interação dos usuários com a função de "pular faixas" no Spotify, torna-se evidente a relevância desse comportamento para a contínua melhoria da experiência do usuário na plataforma.

O site LoudLab[2] explica que taxa de saltos (skip rate) ou contagem de saltos (skip count) é uma métrica usada pela plataforma que indica o número de vezes que um

[2] https://www.loudlab.org/blog/why-is-my-track-getting-skipped-on-spotify/#:~:text=Spotify%20%27skips%20rate%27%20or%20%27,the%20action%20as%20a%20skip

usuário pulou uma música em uma *playlist*. Para que isso seja computado com uma ação de salto, o usuário deve ter ouvido uma faixa por mais de 30 segundos. Também apontam que Daniel Breiholtz, chefe de programas nórdicos e editorial do Spotify, confirma que os editores da plataforma consideram o *ski-prate* ao comprar novas músicas.

A empresa trabalha como uma plataforma de streaming de música e é líder global, oferece acesso a um vasto catálogo de músicas de diferentes gêneros e artistas. Os usuários podem explorar e descobrir novas músicas, criar lista de reprodução personalizadas, seguir artistas e interagir com outros usuários por meio de recursos sociais, e um de seus comportamentos é o de pular faixas. Luden (2021) mostra que a plataforma se consolidou como um dos principais serviços de streaming de música do mundo, permitindo que as pessoas consigam ter acessos a catálogos de músicas em qualquer lugar e a qualquer momento, recriando a forma como as pessoas descobrem, compartilham e consomem música.

No contexto do Spotify, o termo "pulos" refere-se à ação de um usuário de pular uma faixa antes de sua conclusão e passar para uma próxima música. Esse comportamento de pulos dos usuários pode ser influenciado por vários fatores, incluindo preferências musicais individuais, estado de espírito, momento e contexto de audição.

De acordo com Schedl e Rauger (2015), a importância de compreender o comportamento de usuários nas plataformas de streaming de música, afirmando que a análise das interações dos usuários, como o de pular uma música, fornece informações valiosas para os sistemas de recuperação e recomendação de música.

Entender a interação dos usuários com o comportamento de pulos no aplicativo é importante por várias razões. Chen, Li e Ogihara (2012) investigam o comportamento dos usuários em plataformas online de streaming e compartilhamento de músicas em redes sociais e enfatizam a importância de entender o comportamento do “pulo” de faixas como uma forma de compreender a preferência de usuários e interações sociais no contexto de consumo de música. Em primeiro lugar, isso pode fornecer informações valiosas para a plataforma em termos de compreensão das preferências e padrões de audição dos usuários. Essas informações podem ser usadas para melhorar

a recomendação de músicas, personalizar a experiência do usuário e oferecer conteúdo relevante.

Além disso, compreender os padrões de pulos pode ajudar os artistas e a própria indústria musical a entender como seu público interage com seus produtos. Isso pode afetar decisões relacionadas à promoção, lançamento de singles, seleção de faixas para álbuns e até mesmo o processo criativo dos artistas. Celma e Herrera (2008), ressaltam os desafios que aparecem nas pesquisas de sistema de recomendação de músicas, observando que considerar comportamentos como os padrões de saltos de faixas é algo crucial para compreender e construir recomendações musicais relevantes e envolventes.

É importante ressaltar que a interpretação dos dados de pulos deve ser feita com cuidado, levando em consideração outros fatores, como a duração da música, a posição da faixa em uma *playlist* e o contexto de audição do usuário. Além disso, a percepção individual de uma música pode variar amplamente para cada ouvinte, o que significa que os pulos podem não ser necessariamente um indicador preciso do valor ou qualidade de uma determinada faixa.

4 SPOTIFY SEQUENTIAL SKIP PREDICTION CHALLENGE

A tarefa é prever se faixas individuais encontradas em uma sessão de escuta serão ignoradas por um usuário específico. Para isso, são fornecidas informações completas sobre a primeira metade da sessão de escuta do usuário, enquanto a previsão deve ser realizada na segunda metade. Os participantes têm acesso a metadados, bem como descritores acústicos, para todas as faixas encontradas nas sessões de audição.

A saída de uma previsão é uma variável binária para cada faixa na segunda metade da sessão indicando se ela foi ignorada ou não, com 1 indicando que a faixa foi ignorada e 0 indicando que a faixa não foi ignorada. Para este desafio, usamos o campo skip_2 dos logs de sessão como nosso fundamento.

4.1 Base de Dados

A parte pública do conjunto de dados consiste em aproximadamente 130 milhões de sessões de escuta com interações de usuários associadas no serviço Spotify. Além da parte pública do conjunto de dados, aproximadamente 30 milhões de sessões de escuta são usadas para a tabela de classificação do desafio. Para essas sessões de classificação, o participante recebe todos os recursos de interação do usuário para a primeira metade da sessão, mas apenas os IDs da faixa para a segunda metade. No total, os usuários interagiram com quase 4 milhões de faixas durante essas sessões, e o conjunto de dados inclui recursos acústicos e metadados para todas essas faixas

5 XGBOOST

Gomes (2019) explica que o XGBoost é um algoritmo baseado em árvore de decisão e que utiliza de uma estrutura de Gradient Boosting, método que aplica o princípio de impulsionar *weak learners* usando arquitetura de gradiente descendente. Nesse caso o XGBoost aprimora essa estrutura por meio de otimização de sistemas e aprimoramento algorítmicos.

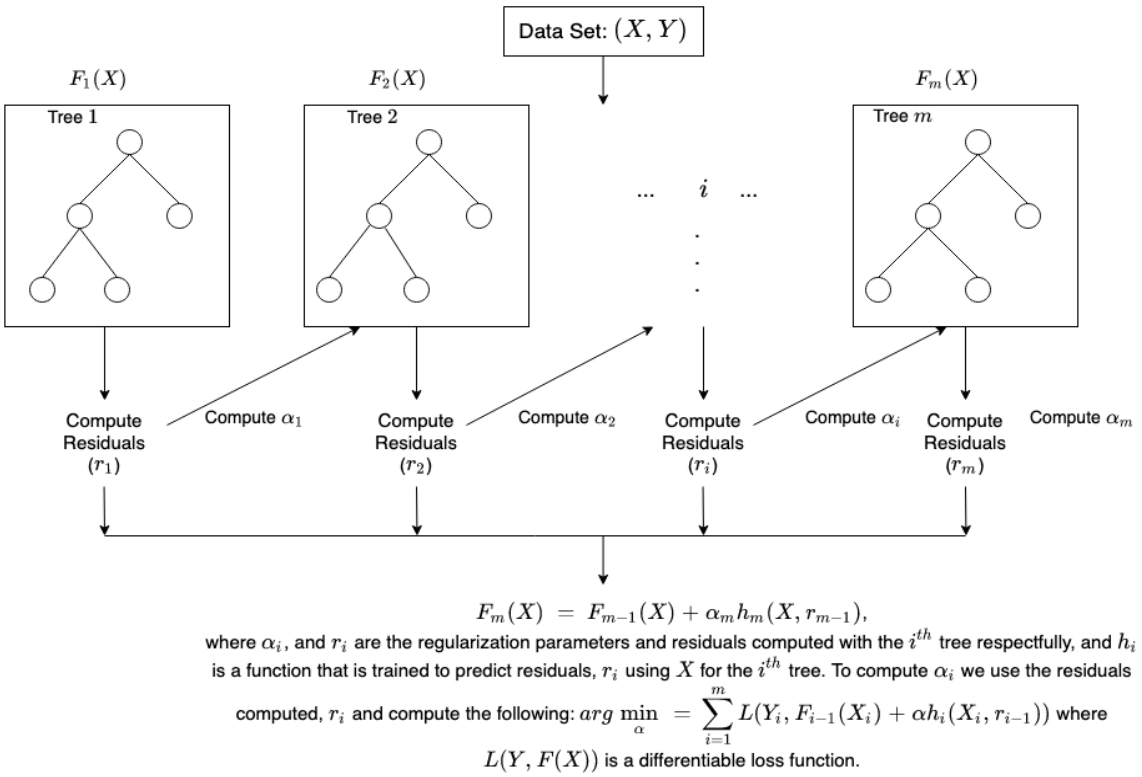
De acordo com Bhattacharya *et al.* (2020). A aplicação do algoritmo ganhou imenso impulso pelas suas implementações em conjuntos de dados tabulares e estruturados. O algoritmo é utilizado amplamente por razão de seu desempenho na modelagem de novos atributos e classificação de rótulos.

Segundo o guia de desenvolvedor da Amazon (Amazon SageMaker Developer Guide):

> Ao usar o aumento de gradiente para regressão, os aprendizes fracos são árvores de regressão e cada árvore de regressão mapeia um ponto de dados de entrada para uma de suas folhas que contém uma pontuação contínua. “O XGBoost minimiza uma função objetivo regularizada (L1 e L2) que combina uma função de perda convexa (com base na diferença entre as saídas preditas e de destino) e um termo de penalidade para a complexidade do modelo (em outras palavras, as funções da árvore de regressão). O treinamento prossegue de forma iterativa, adicionar novas árvores que preveem os resíduos ou erros de árvores anteriores que são então combinadas com árvores anteriores para fazer a previsão final. Chama-se aumento de gradiente porque usa um algoritmo de descida de gradiente para minimizar a perda ao adicionar novos modelos.

Abaixo compreende-se uma ilustração de como o Gradient Tree Boosting funciona:

Figura 1 – Funcionamento do Gradient Tree Boosting.



Fonte: How XGBoost Works – Amazon SageMaker Developer Guide.

Neste contexto, os parâmetros de regularização "a_i" e os resíduos "r_i" são utilizados. Os "a_i" controlam o impacto relativo das árvores individuais no modelo final, evitando o ‘overfitting’. Os "r_i" representam a diferença entre os valores reais e os previstos pelo modelo. A função "h_i" é treinada para prever os "r_i", utilizando "X" como entrada da árvore. Esse traz como resultado um modelo mais preciso.

Chen e Guestrin (2016) apresentam em detalhes os conceitos matemáticos das estruturas do algoritmo, dessa forma é possível demonstrar passo a passo os processos que os autores descrevem, sendo eles:

Primeiro, um método de conjunto de árvores de classificação e árvores de regressão (CARTs) com um conjunto de $K_i E|i \in 1...K\ de$ Nós. A saída de previsão final

do rótulo de classe $\hat{y}_i$ é calculada com base nas pontuações totais de previsão em um nó folha f_k para cada árvore k_{th}:

Figura 2 - Cálculo previsão de classificação de árvores.

$$\hat{y}_i = \varphi(x_i) = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in F,$$

Fonte: Artigo - An investigation of XGBoost-based algorithm for breast cancer classification.

Onde x_i é o conjunto de treinamento e f representa o conjunto de todas as pontuações K para todos os CARTs. Em seguida, uma etapa de regularização é aplicada para melhorar os resultados, conforme mostrado na equação:

Figura 3 - Equação de regularização dos resultados.

$$\mathcal{L}(\varphi) = \sum_i \ell(\hat{y}_i, y_i) + \sum_k \Omega(f_k),$$

Fonte: Artigo - An investigation of XGBoost-based algorithm for breast cancer classification.

Onde "ℓ" representa a função de perda diferencial, definida calculando a diferença de erro entre o alvo y_i e os rótulos de classe previstos $\hat{y}_i$. A segunda parte realiza a penalização Ω na complexidade do modelo para evitar problemas de *overfitting*. A função para a penalidade Ω é calculada por:

Figura 4 – Função de penalidade para evitar *o overfitting*.

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2,$$

Fonte: Artigo - An investigation of XGBoost-based algorithm for breast cancer classification.

Onde γ e λ são parâmetros configuráveis para controlar o grau de regularização. T representa as folhas da árvore e w armazena o valor dos pesos de cada folha.

Em seguida, o Gradient Boosting (GB) é aplicado para resolver efetivamente o problema de classificação junto com a função de perda e estendido por uma segunda expansão de Taylor. O termo constante será removido para obter um objetivo simplificado no passo t, calculado na equação:

Figura 5 - Aplicação do *Gradient Boosting* com uso da expansão de Taylor.

$$\begin{aligned}
\tilde{\mathcal{L}}^{(t)} &= \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \\
&= \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \\
&= \sum_{j=1}^{T} \left[(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2 \right] + \gamma T
\end{aligned}$$

Fonte: Artigo - An investigation of XGBoost-based algorithm for breast cancer classification.

Onde $I_j = \{i | q(x_i) = j\}$ denota a instância da folha t, e a equação para a primeira "g_i" e a segunda ordem "h_i" estatísticas de gradiente da função de perda são definidas nas equações:

Figura 6 – Função de perda do gradiente.

$$g_i = \frac{\partial \ell(\hat{y}_i^{(t-1)}, y_i)}{\partial \hat{y}_i^{(t-1)}}$$

$$h_i = \frac{\partial^2 \ell(\hat{y}_i^{(t-1)}, y_i)}{\partial (\hat{y}_i^{(t-1)})^2}$$

Fonte: Artigo - An investigation of XGBoost-based algorithm for breast cancer classification.

O peso ideal w_j^* da folha j pode então ser calculado pela equação:

Figura 7 – Equação de pesos de cada folha do algoritmo de árvore.

$$w_j^* = \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}$$

Fonte: Artigo - An investigation of XGBoost-based algorithm for breast cancer classification.

Uma função a ser usada como uma característica de pontuação para medir a qualidade de uma estrutura de árvore q, para uma determinada estrutura de árvore $q(xi)$ pode ser calculada pela equação:

Figura 8 – Função de pontuação de qualidades de estrutura de árvore.

$$\tilde{\mathcal{L}}^{(t)}(q) = -\frac{1}{2} \sum_{j=1} T \frac{(\sum_{i \in Ij} g_i)^2}{\sum_{i \in Ij} h_i + \lambda} + \gamma T$$

Fonte: Artigo - An investigation of XGBoost-based algorithm for breast cancer classification.

Normalmente, para medir os nós divididos aplicando pontuação no conjunto de instâncias dos nós esquerdo I_L e direito I_R após a divisão, a redução de perda após a divisão é calculada dessa forma:

Figura 9 – Equação de redução de perda pós divisão de nós.

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{(\sum_{i\in I_L} g_i)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{(\sum_{i\in I_R} g_i)^2}{\sum_{i\in I_R} h_i + \lambda} + \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma$$

Fonte: Artigo - An investigation of XGBoost-based algorithm for breast cancer classification.

Onde: $i = ir \cup il$.

6 ANÁLISE DO BANCO DE DADOS

Essa seção apresenta os processos de obtenção de dados assim como o tratamento e os recursos usados para trabalhar com a base de dados fornecida pelo próprio Spotify.

6.1 Visão geral da base de dados

Utilizou-se uma versão minimizada do banco de dados original, devido a limitação de poder de processamento. O documento “Training_Set_And_Track_Features_Mini” é disponibilizado pelo próprio Spotify[3].

O arquivo está compactado em formato “tar.gz” e consiste em dois DataFrames, um nomeado como track_features e o outro training_set.

[3] https://www.aicrowd.com/enges/spotify-sequential-skip-prediction-challenge/dataset_files.

A maioria das informações são variáveis de ponto flutuante representados por uma mantissa, que contém o valor numérico e um expoente contendo a ordem de grandeza do número (Whitney *et al.*, 2023). Alguns valores são números inteiros e outros são objetos textuais (sequências de caracteres alfanuméricos (letras, números e/ou símbolos), caracterizando valores categóricos. Segundo Moffit (2018), um object é considerado uma string na biblioteca pandas, então performa como valores textuais ao invés de matemáticos.

Observar-se que além dos valores inteiros, contínuos e textuais temos valores booleanos, de acordo com Whitney[4] et al. (2023) “bool” é uma variável que pode ter valores “Verdadeiros” (True) ou “Falso” (False), ou valores binários, “um ou outro”.

6.2 Métodos de exploração dos dados

Para adquirir uma compreensão aprofundada dos dados processados e desenvolver modelos de Machine Learning para previsões, utilizou-se a linguagem de programação Python em conjunto de diversificadas bibliotecas, incluindo Pandas, Numpy, Seaborn e Matplotlib.

Criada por Guido Van Hossum, a linguagem Python tinha o objetivo de ser uma tecnologia fácil e intuitiva, para poder resolver os problemas da extensão e a complexidade da linguagem C, que apenas programadores experientes tinham noção de entender o funcionamento em determinados softwares. (Silva; Silva, 2019).

Em Python, as bibliotecas são módulos, estes são arquivos com a extensão “.py”, contendo códigos Python que podem ser importados dentro de outro Programa Python. De forma mais simples, é uma biblioteca de arquivos que contém conjuntos de funções que o usuário deseja incluir em seu aplicativo. (Goyal, 2023)

6.3 Base de dados das características de trilhas

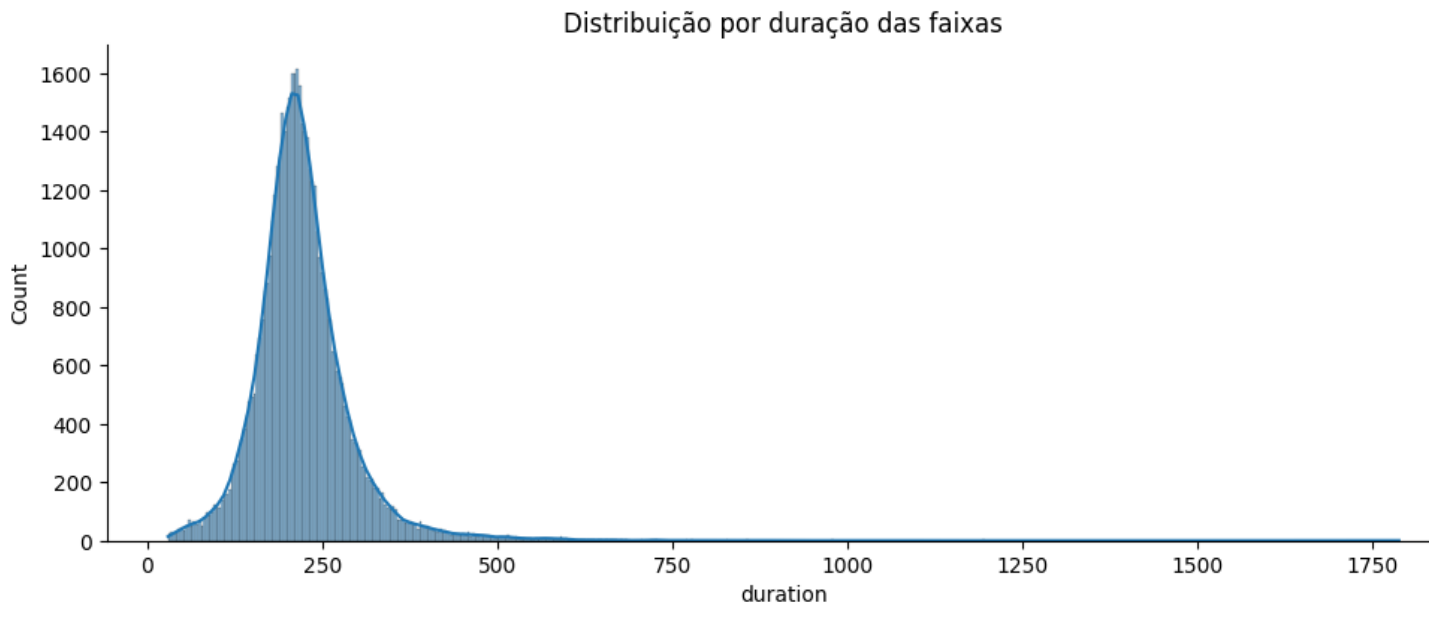
[4] https://learn.microsoft.com/pt-br/cpp/cpp/bool-cpp?view=msvc-170

Com o auxílio de métodos estatísticos, os valores totais de dados, as médias de cada coluna, juntamente com o cálculo do desvio padrão, os valores mínimo e máximo, além dos quartis para cada coluna do DataFrame foram extraídos. Não foram identificados valores atípicos, como datas negativas ou discrepâncias significativas entre os valores.

Com a utilização das bibliotecas "Seaborn" e "Matplotlib", possibilitou-se o avanço na exploração dos dados, com o objetivo de compreender a distribuição de faixas.

Observa-se que a maioria das faixas apresenta duração distribuídas entre 150 e 300 segundos, com um pico significativo de 250 segundos de tempo das músicas demonstrado no gráfico 10.

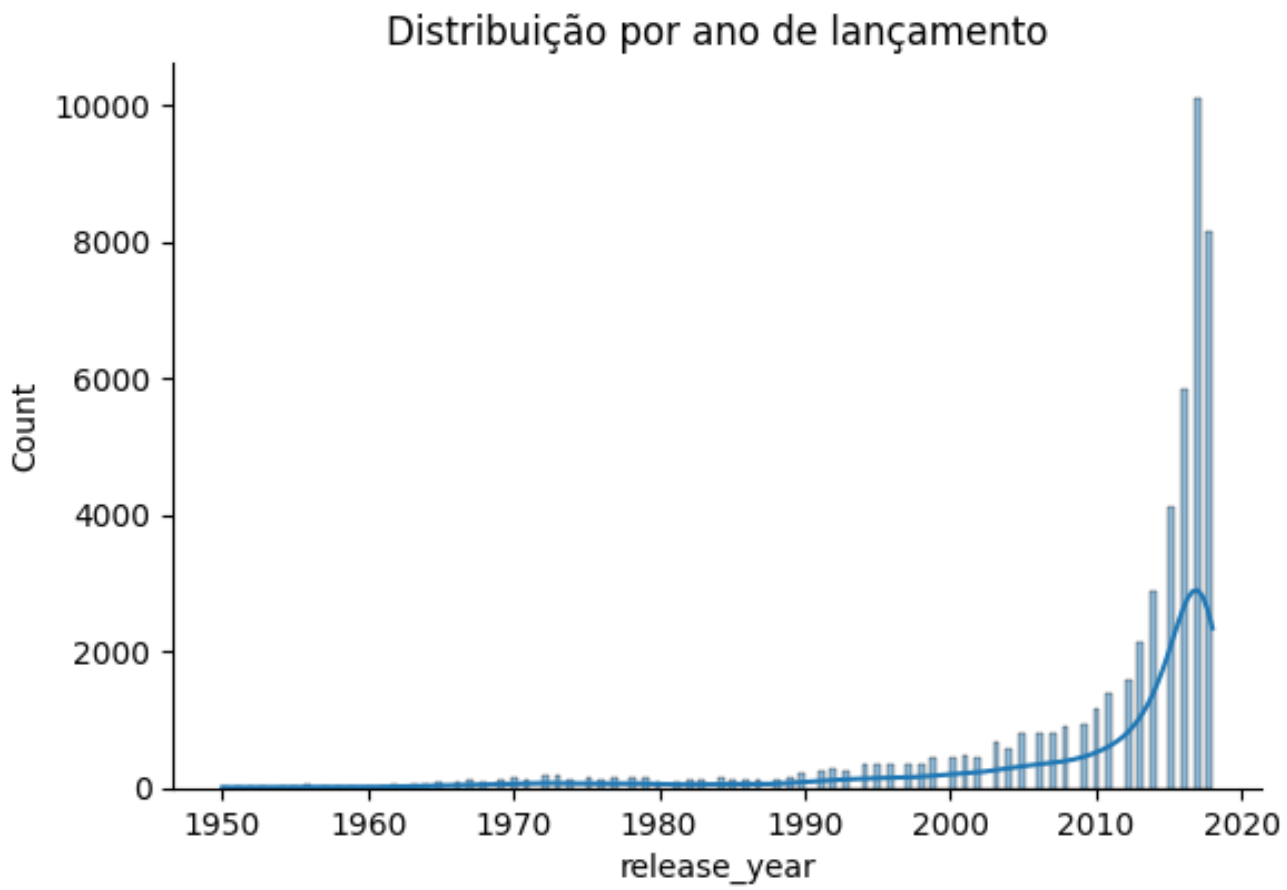
Figura 10 – Distribuição de músicas por duração das faixas



Fonte: Elaboração Própria.

Além disso, constata-se que a maior parte das faixas da base de dados utilizada foi lançada no período compreendido entre os anos de 2010 e 2020 conforme apresentado no gráfico abaixo (figura 11).

Figura 11 – Distribuição de músicas por ano de lançamento

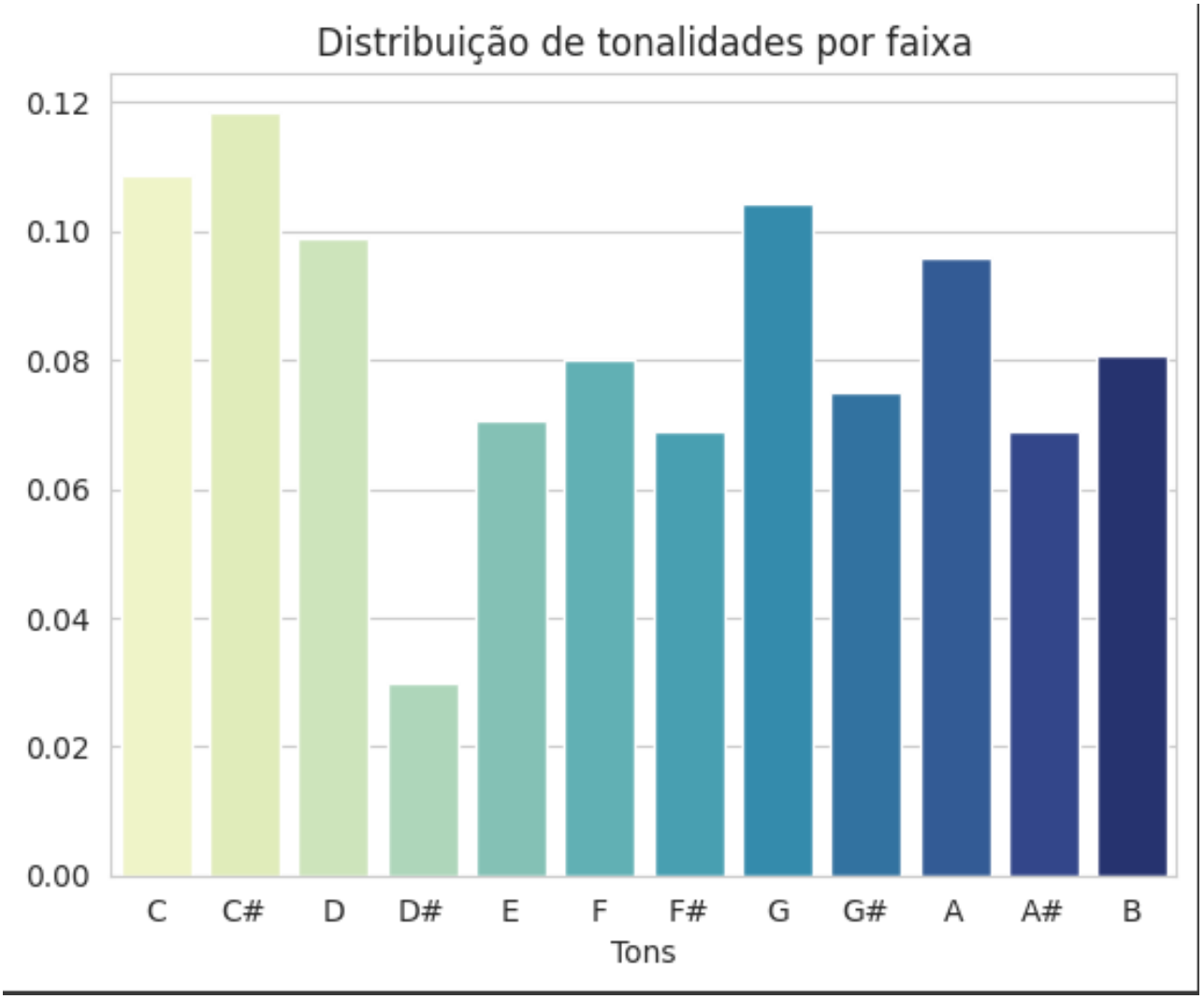


Fonte: Elaboração Própria.

Outro fator notável são as distribuições das escalas musicais e tonalidades. Mais de 64% das músicas estão em escala maior, enquanto aproximadamente 35% estão em escala menor. Essa informação sugere uma predominância de músicas com tonalidades mais alegres e otimistas, representadas pela escala maior, em comparação às tonalidades mais melancólicas e introspectivas, representadas pela escala menor.

Constata-se uma predominância de 32.555 ocorrências de músicas em escala maior em comparação a 18.149 ocorrências em escala menor. Além disso, foi possível identificar a distribuição dos tons presentes em cada faixa, revelando uma predominância significativa dos tons de Dó Sustenido/Bemol, Dó e Sol, conforme demonstrado na Figura 12. Essas descobertas fornecem informações relevantes para análise e compreensão das características musicais presentes na base de dados, contribuindo para a investigação em questão.

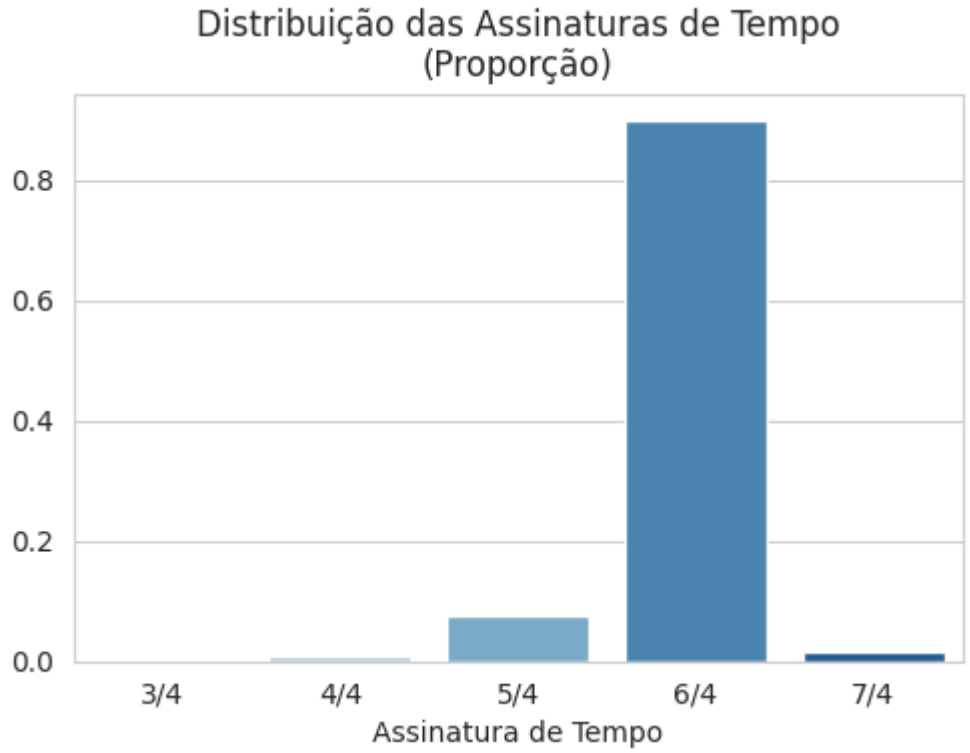
Figura 12 – Distribuição de tonalidade musical por faixas.

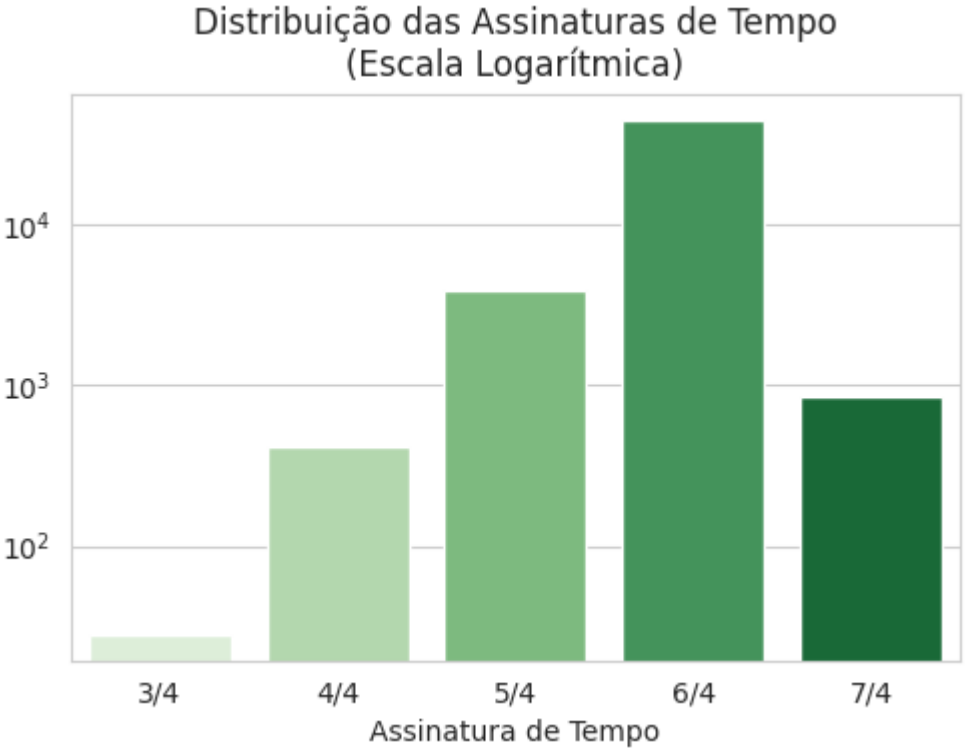


Fonte: Elaboração Própria.

Uma descoberta notável reside na distribuição das assinaturas de tempo encontradas no estudo. Enquanto é comum encontrar assinaturas de 4/4 nas músicas mais populares, os resultados revelaram uma predominância significativa de compassos de 6/4, seguidos por compassos de 5/4. Essa distribuição pode ser observada de forma mais clara na Figura 13.

Figura 13 – Distribuição de assinaturas de tempo em proporção e escala logarítmica.





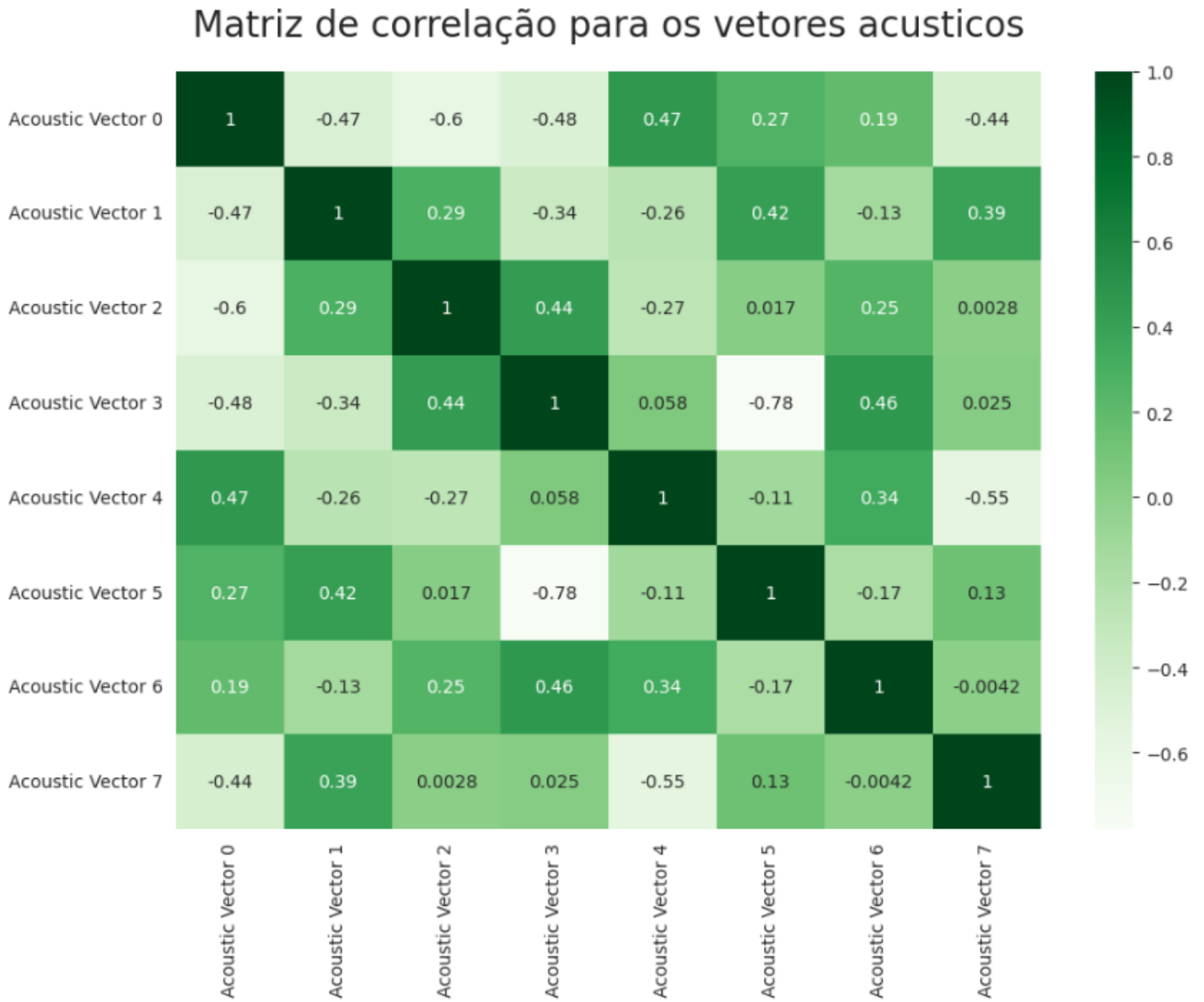
Fonte: Elaboração Própria.

Com o intuito de visualizar adequadamente os dados, foi adotada uma escala logarítmica na base "y" do gráfico. Essa escolha se justifica pela ampla gama de valores

observados, os quais apresentavam discrepâncias significativas entre si. A distribuição logarítmica permite que os valores sejam dispostos de acordo com os expoentes de uma base numérica específica. Essa abordagem implica que as distâncias entre os valores na escala logarítmica não são uniformes, tornando as diferenças significativas entre os valores mais facilmente discerníveis.

Por fim, foi realizada uma análise da correlação dos vetores acústicos utilizando o método de correlação de Pearson, que nos fornece uma tabela de correlação. O coeficiente de correlação de Pearson é uma medida estatística que avalia a relação linear entre duas variáveis, indicando a intensidade e a direção dessa relação (Oliveira, 2019). Essa análise permitiu avaliar a associação entre os diferentes vetores acústicos e identificar padrões de dependência ou independência entre eles. A tabela de correlação de Pearson apresenta os valores de correlação para cada par de vetores acústicos, proporcionando uma visão mais precisa das relações existentes entre as variáveis estudadas.

Figura 14 – Matriz de correlação entre os vetores acústicos.



Fonte: Elaboração Própria.

Os coeficientes de correlação variam de -1 a 1, e valores próximos a 1 indicam uma correlação positiva forte, indicando uma relação direta onde o aumento em uma variável é acompanhado pelo aumento na outra. Valores próximos a -1 representam uma correlação negativa forte, indicando uma relação inversa, onde o aumento em uma variável está associado à diminuição na outra. Por outro lado, valores próximos a 0 indicam uma ausência de correlação, sugerindo a falta de uma relação linear significativa entre as variáveis estudadas.

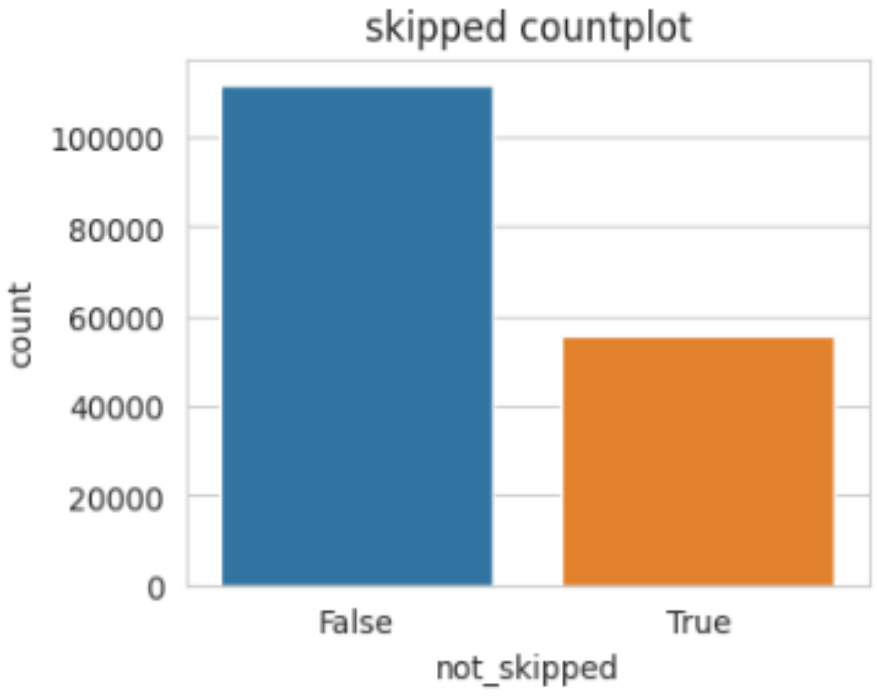
As correlações entre os vetores acústicos não apresentaram valores fortemente positivos, com a maioria se aproximando de 0,5. A correlação mais alta foi de

0,47 entre o vetor acústico 4 e o vetor 1. Observou-se uma correlação negativa alta de -0,78 entre o vetor acústico 5 e o vetor 3. Essas correlações podem influenciar a análise e interpretação dos resultados, sendo uma característica relevante para o algoritmo de predição.

6.4 Base de dados das características dos usuários

Ao analisar o DataFrame inicialmente, pode-se observar que ele possui 167.880 linhas e 21 colunas. Sua amplitude é consistentemente maior do que a base de trilhas musicais, pois engloba dados de diversos usuários e seus respectivos comportamentos. Ao realizar uma análise do conjunto de dados, é possível observar uma diferença significativa entre as faixas que foram puladas (*true*) e as que não foram puladas (false), como ilustrado no gráfico apresentado. Essa disparidade evidencia a existência de um padrão distinto entre as faixas que foram puladas e as que não foram puladas, o que pode ser relevante para a compreensão do comportamento dos dados.

Figura 15 – Diferença de pulos por total de usuários.

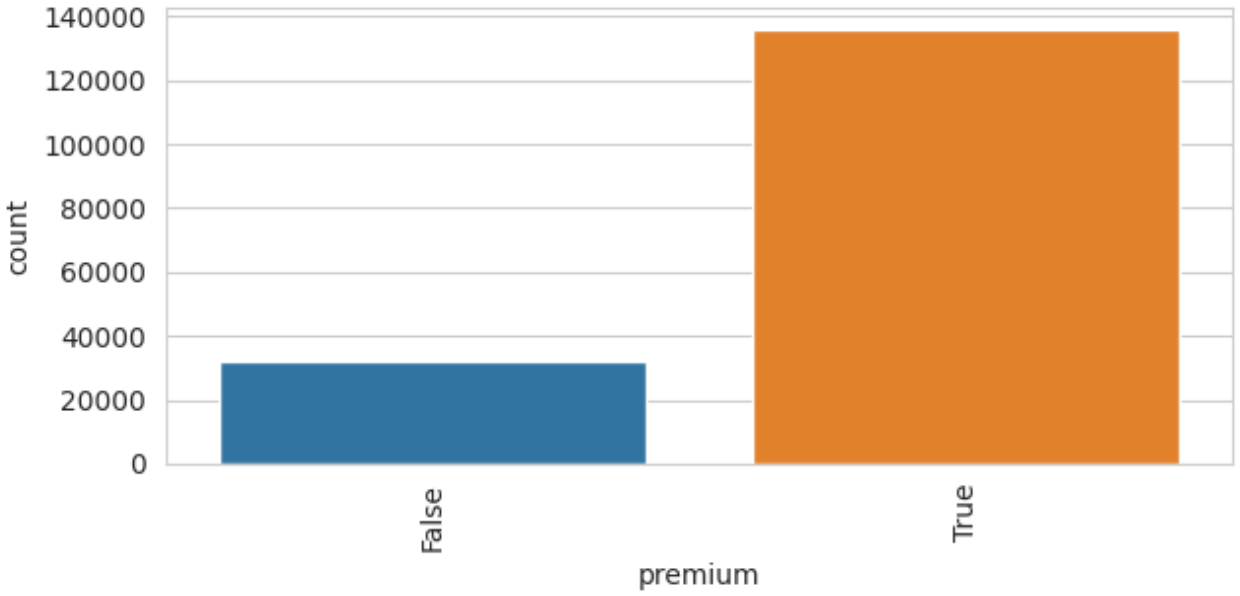


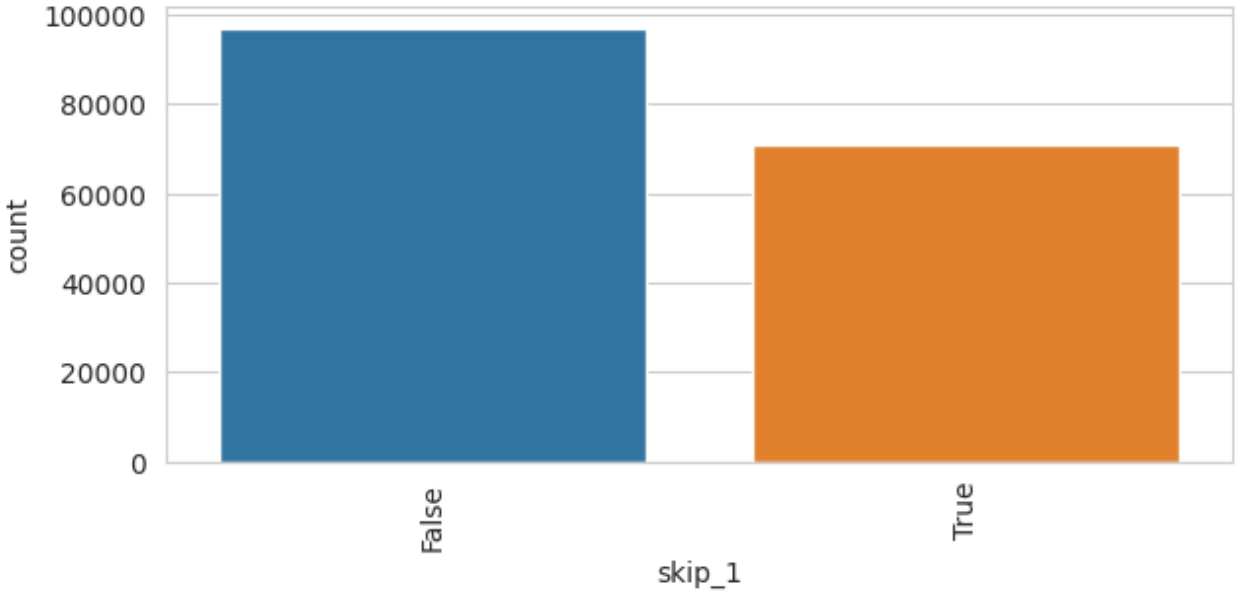
Fonte: Elaboração Própria.

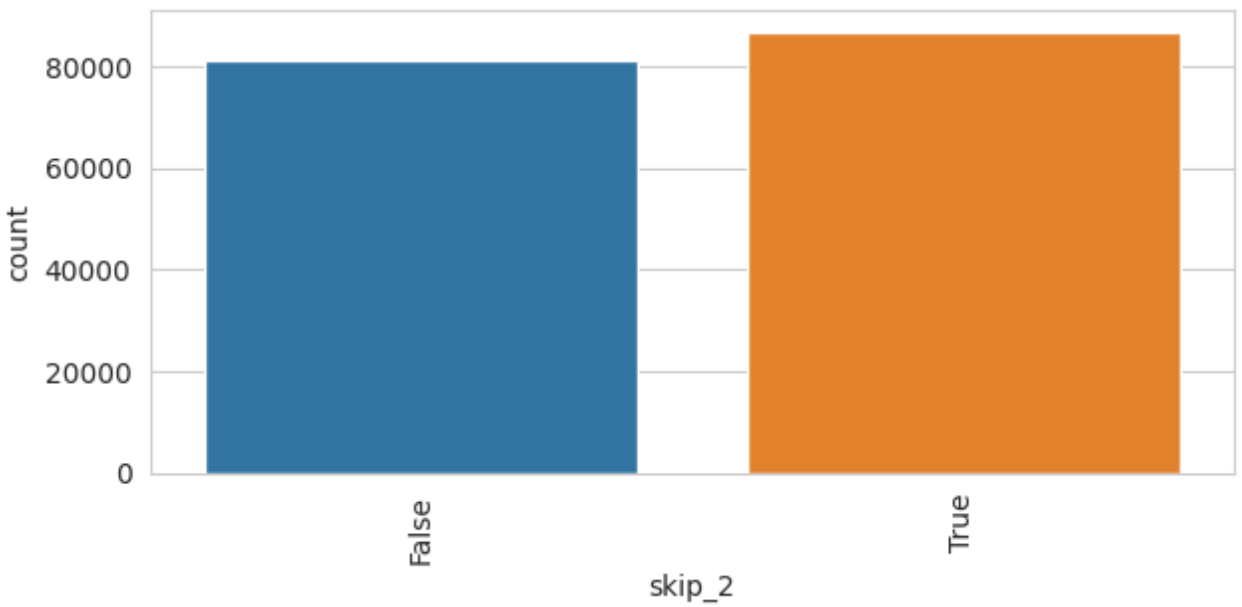
Foram realizados cálculos para determinar a porcentagem de diferença entre as faixas puladas e não puladas. Constatou-se que 66,71% das faixas não foram puladas pelos usuários, enquanto 33,29% foram puladas. Esses números indicam uma diferença significativa entre as duas categorias de faixas em termos de preferência dos usuários.

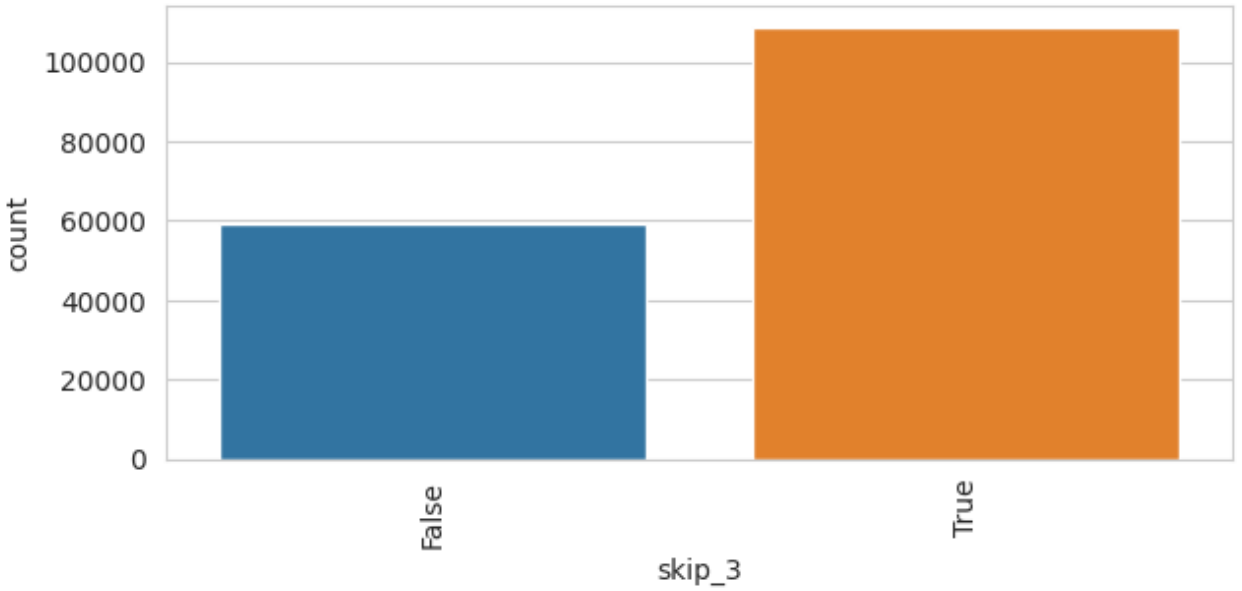
Embora o Dataframe seja de maior tamanho, foi possível realizar uma análise concisa dos dados, incluindo a contagem de várias características. A figura a seguir ilustra os resultados obtidos:

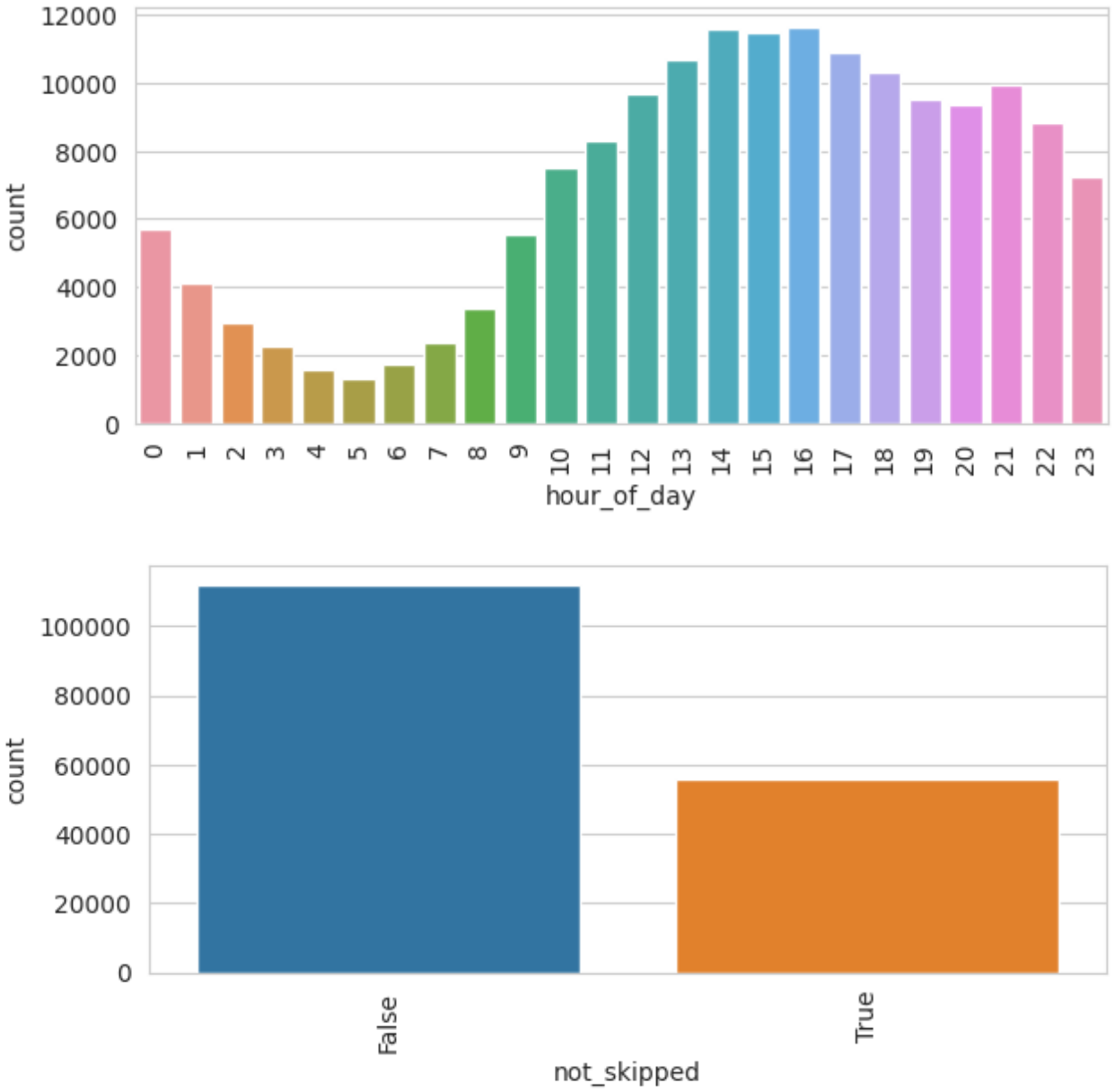
Figura 16 - Contagens de características dataframe training_set.


140000
120000
100000
80000
60000
40000
20000
0
count
False
True
premium


100000
80000
60000
40000
20000
0
count
False
True
skip_1


80000
60000
40000
20000
0
count
False
True
skip_2


100000
80000
60000
40000
20000
0
count
False
True
skip_3



Fonte: Elaboração Própria.

Na análise, foram contabilizadas diversas características do conjunto de dados, fornecendo informações valiosas sobre cada uma delas. Essa abordagem permitiu uma compreensão mais aprofundada dos padrões e tendências presentes nos dados, facilitando a extração de dados relevantes.

Durante a análise, foi possível identificar várias tendências e padrões nos dados. Primeiramente, constatou-se que a maioria dos usuários possui uma assinatura

premium. Isso sugere que a plataforma de música é popular entre os usuários dispostos a pagar por recursos adicionais.

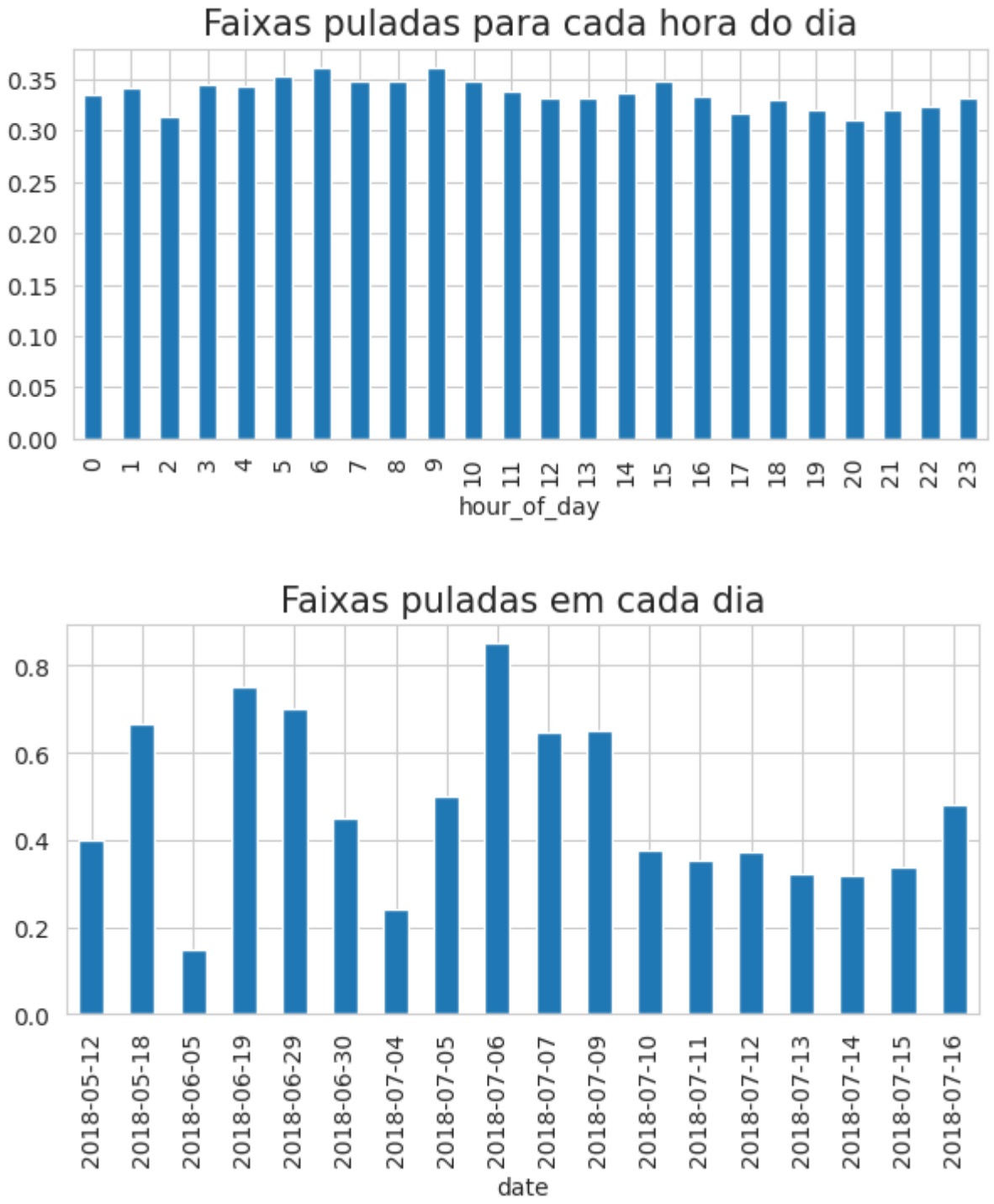
Ao examinar o gráfico superior direito, que indica os momentos em que os usuários pulam nas faixas musicais, observa-se que a maioria dos pulos ocorre no primeiro terço da música. Os pulos estão representados em azul (*false*) e as faixas não puladas em laranja (*true*). Essa análise revela que uma a maioria das músicas que são ignoradas, acontecem nos seus primeiros momentos.

O gráfico inferior esquerdo ilustra os horários de uso do aplicativo, revelando padrões de comportamento dos usuários. Foi observado que o número de usuários começa a aumentar a partir das 9 horas da manhã, atingindo picos nos horários das 14h e 16h. Em seguida, o número de usuários começa a diminuir, atingindo seu valor mínimo às 5h da madrugada. Essa análise temporal dos usuários fornece insights sobre os momentos de maior atividade e engajamento com o aplicativo ao longo do dia.

Outros dados relevantes que foram identificados incluem a distribuição das faixas puladas ao longo das horas do dia e em cada dia específico. Observou-se que a distribuição das faixas puladas mantém um padrão em relação aos diferentes horários do dia, com um pico ocorrendo às 2 horas. É importante lembrar que estamos comparando esses dados com a coluna "*not_skipped*", onde os valores falsos representam as faixas que foram puladas.

Além disso, foi constatado que a maioria das faixas puladas ocorreu no dia 6 de maio de 2018, conforme demonstrado no gráfico abaixo:

Figura 17 – Total de faixas puladas por dia e hora do dia.



Fonte: Elaboração Própria.

Essas informações indicam que o horário das 2 horas é um período em que ocorre uma quantidade significativa de faixas puladas pelos usuários. Além disso, o dia 6 de maio de 2018 se destaca como o dia com a maior incidência de faixas puladas.

Esses insights contribuem para uma compreensão mais profunda dos padrões de comportamento dos usuários em relação às faixas puladas, tanto em termos de horários específicos quanto de dias específicos.

7 TRATAMENTO DOS DADOS E MODELOS DE PREDIÇÃO

Para a construção dos modelos, realizou-se o pré-processamento dos dados a fim de otimiza-los para uso. Inicialmente, as duas tabelas foram combinadas utilizando os campos "*track_id*" e "*track_id_clean*" como chave de união. Como resultado, obteve-se um novo DataFrame contendo 167.880 linhas e 51 colunas. Essa fusão permitiu agregar as informações das tabelas originais em uma única estrutura de dados, facilitando análises e modelagens futuras. Essa etapa de unificação das tabelas é essencial para garantir que todas as informações relevantes sejam consideradas de maneira conjunta.

No contexto deste trabalho, considerando o campo "skip_2" como a fonte verdadeira dos pulos de faixas dos usuários, foi implementada a criação da coluna denominada "skipped". Essa coluna é preenchida com valores booleanos, onde 1 representa a ocorrência de um pulo de faixa (verdadeiro) e 0 indica a ausência de pulo de faixa (falso). Para determinar a presença ou ausência de pulos de faixa, verifica-se os valores das colunas "skip_1" e "skip_2". Se algum desses campos possuir o valor verdadeiro (1), então a coluna "skipped" é preenchida com o valor verdadeiro (1). Caso contrário, se ambas as colunas "skip_1" e "skip_2" apresentarem o valor falso (0), a coluna recebe o valor falso (0). Essa abordagem permite consolidar as informações dos pulos de faixa dos usuários em uma única coluna, simplificando a análise posterior dos dados relacionados a essa métrica, possibilitando o uso como um vetor de saída para o modelo de predição.

Segundo Shaikh (2018), a codificação de recursos categóricos em valores numéricos pode ser facilmente realizada utilizando o método *LabelEncoder* na linguagem de programação Python, por meio da biblioteca Sklearn. Essa abordagem demonstra ser uma ferramenta altamente eficiente na tarefa de transformar níveis de

recursos categóricos em representações numéricas. O *LabelEncoder* atribui valores numéricos aos rótulos, variando de 0 a n_classes-1, onde n representa o número de rótulos distintos presentes no conjunto de dados. Vale ressaltar que, caso um rótulo seja repetido, o mesmo valor numérico atribuído anteriormente será mantido. Com base nesse procedimento, é possível converter todas as colunas categóricas do conjunto de dados em valores numéricos de forma consistente e coerente.

A fim de obter insights sobre a importância de cada coluna do DataFrame, um modelo de classificação usando o algoritmo de Árvores Aleatórias (Random Forest) foi usado. O Random Forest é um algoritmo supervisionado de Aprendizado de Máquina amplamente utilizado para problemas de classificação e regressão. Uma das suas qualidades é a capacidade de identificar os recursos mais importantes em uma base de dados (Saini, 2021).

A importância de cada coluna é calculada usando a fórmula matemática baseada na redução da impureza, conhecida como "*Gini Importance*", em um Random Forest. Essa fórmula leva em consideração como cada coluna contribui para a redução da impureza ao realizar as divisões nas árvores do modelo.

Figura 18 – Equação da importância de cada feature da árvore de decisão.

$$fi_i = \frac{\sum_{j:node\ j\ splits\ on\ feature\ i} ni_j}{\sum_{k \in all\ nodes} ni_k}$$

Fonte: RONAGHAN. S; Towards Data Science

Onde:

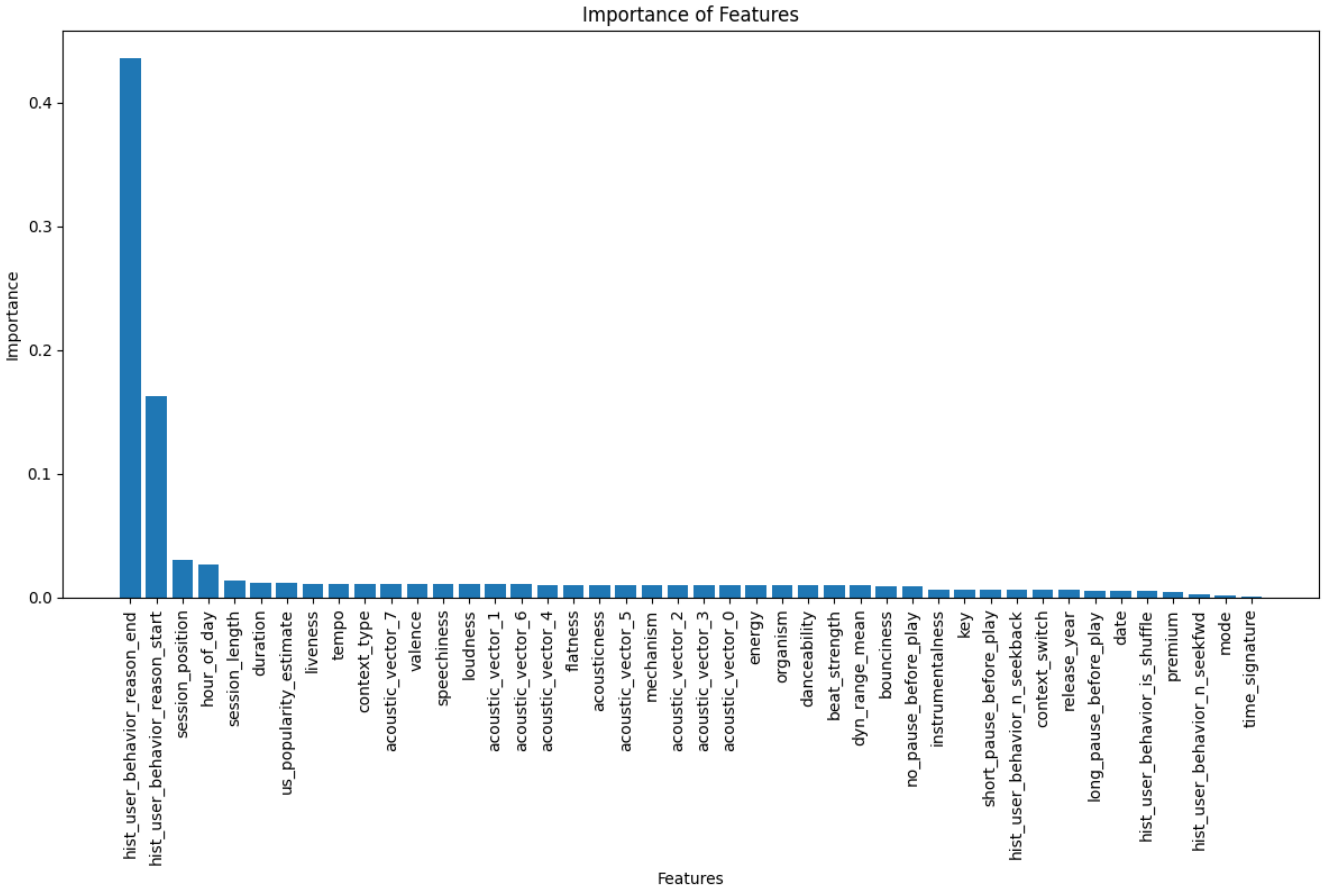
fi_i sub (i) = a importância do recurso i^{ni}

sub (j) = a importância do nó j

Gini Importance = Soma (Redução da impureza de cada nó dividida pelo número total de nós) / Número de árvores.

Para cada variável em um Random Forest, a importância é calculada através da média ponderada da redução da impureza que a variável traz para cada nó de decisão em cada árvore. (Ronaghan, 2018). Com o modelo aplicado, obtém-se a sua visualização gráfica:

Figura 19 – Extração dos elementos mais importantes.



Fonte: Elaboração Própria.

Outro método usado para entender a importância de cada feature foi valores de SHAP. De acordo com Liu et al. A estrutura SHAP é baseada no cálculo do valor de Shapley para medir como os recursos afetam a variável dependente. Na tarefa de seleção de recursos, o valor SHAP pode ser utilizado para calcular a contribuição marginal de cada recurso e medir a importância dele. Primeiramente, o conjunto de dados original é inserido no modelo, e a estrutura atribui um valor SHAP a cada recurso de cada ponto de dados, representando a contribuição correspondente à predição do modelo. Portanto, o cálculo do valor SHAP depende do modelo. O valor SHAP /j do recurso j é definido como:

Figura 20 – Equação do valor SHAP

$$\phi_j = \frac{1}{|N|!} \sum_{S \subseteq Nleft\{j\}} |S|!(|N| - |S| - 1)![f(S \cup \{j\}) - f(S)]$$

Fonte: LIU et al. Diagnosis of Parkinson's disease based on SHAP value feature selection.

Onde $|\cdot|$ representa o número de elementos no conjunto. N representa o conjunto original de recursos. S representa qualquer subconjunto de recursos em $N. Nleft\{j\}$ representa um subconjunto de todos os elementos na sequência antes do recurso j. $f(S)$ representa a saída do modelo de aprendizado de máquina para o subconjunto de recursos $S. f\ (S \cup \{j\}) - f(S)$ representa o valor cumulativo de contribuição do recurso j.

Na equação (Figura 20), pode-se observar que o valor SHAP /j do recurso j é calculado pela média das contribuições de todas as permutações possíveis do conjunto de recursos. Devido à natureza objetiva da distribuição dos benefícios do valor SHAP e à sua capacidade de medir adequadamente o efeito de cada recurso na previsão ou classificação do modelo, é viável utilizar o valor SHAP como uma medida da importância dos recursos, dessa forma obtemos os resultados das importâncias de cada *feature*:

Com base nos resultados obtidos dos algoritmos de importância de características, identificou-se as variáveis estatisticamente menos relevantes para os modelos possibilitando o descarte.

Uma análise dos balanceamentos dos dados alvos foi realizada, dando um foco no vetor de saída que traz valores binários, sendo 0 valores que indicam que o usuário não pulou a faixa em que estava escutando e 1 para usuários que pularam as faixas, foi observado os seguintes valores:

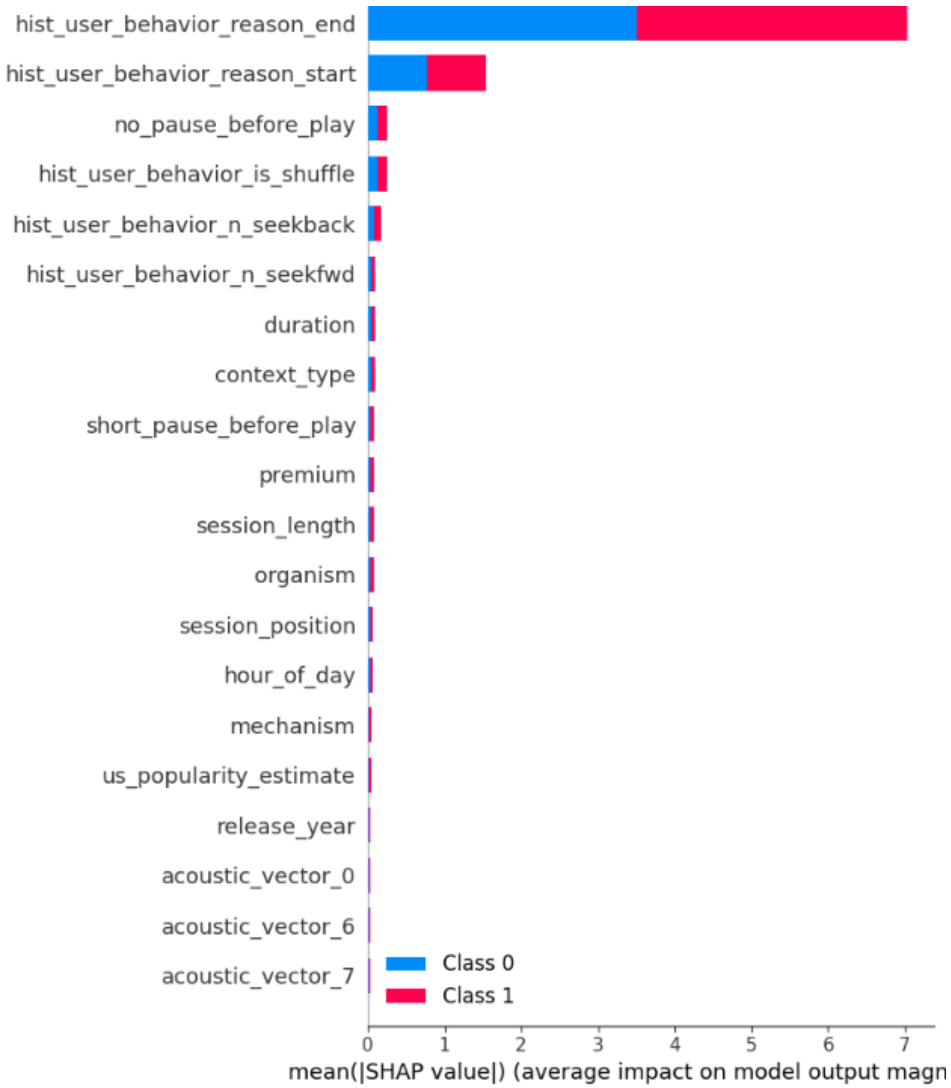
Das 167.880 observações representadas no vetor, observou-se que a ocorrência do valor 1 foi de 86.824 vezes, enquanto o valor 0 foi registrado em 81.056 ocasiões. Tendo um balanceamento percentual de 51,72% para o valor 1 e 48.28% para o valor 0. Sendo uma diferença de 3,44%, isso indica que a base de dados alvo tem um

bom balanceamento de observações, o que diminui a probabilidade de um possível overfitting e enviesamento dos dados nos treinamentos dos algoritmos.

Após a aplicação dos devidos tratamentos na base de dados, foi dado início à construção dos modelos de Aprendizado de Máquina para realizar as predições. Um pipeline foi desenvolvido, contendo sete diferentes modelos de classificação: Regressão Logística, Árvore de Decisão, Floresta Aleatória, *Gradient Boosting*, Máquina de Vetores de Suporte (SVM), *Naive Bayes* Gaussiano e XGBoost.

Adicionalmente, um algoritmo de aprendizado profundo foi modelado utilizando uma Rede Neural Convolucional (CNN) para treinamento e teste dos dados.

Figura 21 – Importância de *features* calculados por valores SHAP.



Fonte: Produção própria.

8 RESULTADOS

Após o processo de treinamento de todos os modelos, constatou-se que o XGBClassifier obteve o desempenho superior em relação aos demais. Embora todos os modelos tenham alcançado acurácia próxima uma das outras, variando entre 83% e 87%, o XGBoost demonstrou resultados ligeiramente superiores em outras métricas avaliadas.

Dessa forma é possível observar os resultados na figura abaixo:

Figura 22 – XGBClassifier, relatório de classificação e métricas.

XGBoost Classification Report				
	Precision	***Recall***	***f1-score***	***Support***
1 (*True*)	0.91	0.82	0.86	16239
0 (*False*)	0.85	0.92	0.88	17337
Accuracy			0.87	33576
Macro avg	0.88	0.87	0.87	33576
Weighted avg	0.88	0.87	0.87	33576

Fonte: Elaboração Própria.

Ao interpretar os resultados de um relatório de classificação como o apresentado acima, é importante entender o significado de cada métrica.

Grandini, Bagli e Visani(2020), oferecem uma análise abrangente das métricas, descrevendo suas características matemáticas distintivas, sendo elas:

A precisão, definida como a razão entre os verdadeiros positivos e o total de unidades classificadas como positivas pelo modelo (soma dos positivos previstos). Os verdadeiros positivos correspondem aos elementos corretamente identificados como positivos pelo modelo, enquanto os falsos positivos referem-se aos elementos incorretamente classificados como positivos, mas que na realidade são negativos.

Figura 23 – Equação da métrica Precisão.

$$Precision = \frac{TP}{TP + FP}$$

Fonte: Gradini, Bagli e Visani – Metrics for multi-class classification: An Overview.

Em resumo, a precisão é uma medida que nos permite avaliar a confiabilidade do modelo na identificação correta de casos positivos.

O Recall, medida que avalia a proporção de elementos corretamente identificados como verdadeiros positivos em relação ao total de unidades que foram classificadas como positivas (soma dos verdadeiros positivos e falsos negativos). Em

outras palavras, os falsos negativos são casos em que o modelo rotulou erroneamente os elementos como negativos, quando, na realidade, eles são positivos.

Figura 24 – Equação da métrica Recall.

$$Recall = \frac{TP}{TP + FN}$$

Fonte: Gradini, Bagli e Visani – Metrics for multi-class classification: An Overview.

O Recall mede a precisão preditiva do modelo para a classe positiva: intuitivamente, ele mede a capacidade do modelo de encontrar todas as unidades positivas no conjunto de dados.

Outras métricas importantes são a acurácia e o F1-Score, também descritas por Grandini, Bagli e Visani (2020), em suas formas matemáticas:

Segundo os autores, acurácia é uma das métricas mais populares para classificação de multi-classes também usada em classificação binária:

Figura 25 – Equação da métrica Acurácia.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Fonte: Gradini, Bagli e Visani – Metrics for multi-class classification: An Overview.

A Precisão é calculada considerando os verdadeiros positivos e verdadeiros negativos no numerador, e todos os elementos da matriz de confusão no denominador. Em resumo, a precisão é a probabilidade de que a previsão do modelo esteja correta.

F1-Score agrega os valores de Precision e Recall sob o conceito da média harmônica.

Figura 26 – Equação da métrica F1-Score.

$$\text{F1-Score} = \left(\frac{2}{precision^{-1} + recall^{-1}} \right) = 2 \cdot \left(\frac{precision \cdot recall}{precision + recall} \right)$$

Fonte: Gradini, Bagli e Visani – Metrics for multi-class classification: An Overview.

O F1-score é uma métrica que representa uma média ponderada entre a Precisão e o Recall. Seu valor varia de 0 a 1, sendo 1 o melhor desempenho e 0 o pior. O F1-score equilibra igualmente a contribuição da Precisão e do Recall, usando a média harmônica. Isso é útil para encontrar o melhor equilíbrio entre essas duas medidas.

Em casos binários, usa-se Precisão e Recall. Parâmetro confiável para identificar pontos fracos do algoritmo de previsão, se houver algum. É uma métrica útil para avaliar o desempenho geral do modelo, considerando a precisão das previsões e a capacidade de recuperar instâncias positivas corretamente.

Essas consistem nas principais métricas para a análise dos resultados do modelo matemático selecionado.

No caso do relatório de classificação fornecido para o algoritmo XGBoost, temos informações sobre as classes "0" e "1":

Para a classe "0":

- A precisão é de 0.91, o que significa que 91% das classificações feitas pelo modelo para esta classe estão corretas.
- O recall é de 0.82, o que indica que o modelo conseguiu encontrar corretamente 82% das amostras positivas desta classe.
- O F1-score é de 0.86, que é uma média harmônica da precisão e do recall. É uma medida geral de desempenho que leva em consideração tanto a precisão quanto o recall.
- O suporte é de 16239, o que representa o número de amostras da classe "0" no conjunto de dados.

Para a classe "1":

- A precisão é de 0.85, indicando que 85% das classificações feitas pelo modelo para esta classe estão corretas.

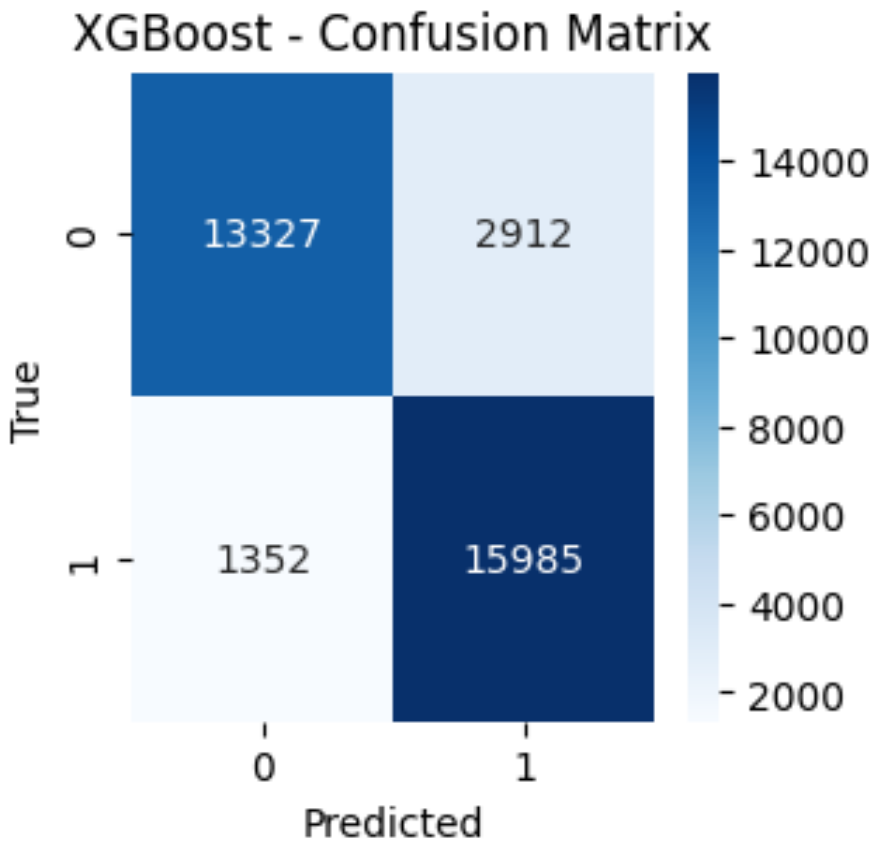
- O recall é de 0.92, o que significa que o modelo conseguiu encontrar corretamente 92% das amostras positivas desta classe.
- O F1-score é de 0.88, uma medida geral de desempenho para esta classe.
- O suporte é de 17337, o número de amostras da classe "1" no conjunto de dados.

No geral, os resultados indicam um bom desempenho do modelo. A acurácia geral do modelo é de 0.87, o que significa que ele classificou corretamente 87% das amostras no conjunto de dados. A média ponderada da precisão (weighted avg precision), recall (weighted avg recall) e F1-score (weighted avg f1-score) é de 0.88, o que indica que o desempenho médio do modelo em todas as classes foi alto.

Quanto à matriz de confusão, podemos analisar o desempenho do modelo em relação às classes 0 e 1.

A matriz de confusão é uma ferramenta para avaliar o desempenho de um classificador em tarefas de classificação binária. Ela consiste em uma matriz quadrada que mostra as proporções de instâncias classificadas corretamente e incorretamente. Na matriz, a classe 1 é representada por "P" e a segunda classe (ou todas as classes diferentes da classe 1 em um cenário multi-classe) é representada por "N" (Raschka, 2014).

Figura 27 – Matriz de confusão



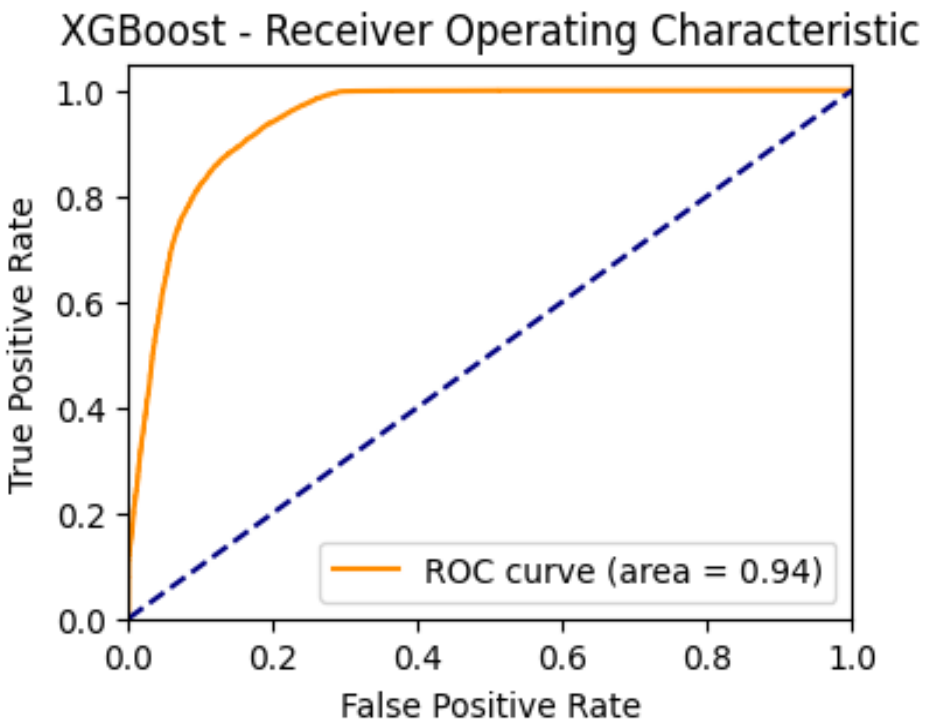
Fonte: Elaboração Própria.

Para classe 0 o modelo classificou corretamente 13.327 valores como pertencentes a classe 0 (verdadeiro negativo) e classificou de forma incorreta 2.912 como pertencentes a classe 1 (falsos positivos).

Para classe 1 foi classificado corretamente 15.985 instâncias como pertencentes a classe 1 (verdadeiro positivo) e erroneamente classificou 1.352 instâncias como pertencentes a classe 0 (falsos negativos).

Outra métrica usada para entender a eficiência do modelo foi o gráfico ROC. Segundo Raschka (2014), o Receiver Operator Characteristics (ROC) é útil para avaliar o desempenho de modelos de classificação em termos das taxas de falso positivo e verdadeiro positivo. Esses gráficos ajudam na seleção de modelos mais adequados para a tarefa de classificação, como apresentado abaixo:

Figura 28 – Gráfico ROC – XGBoost Classifier.



Fonte: Elaboração Própria.

A diagonal de um gráfico ROC representa a suposição aleatória, enquanto os modelos de classificação abaixo da diagonal são considerados piores do que suposições aleatórias. Um classificador perfeito seria representado no canto superior esquerdo do gráfico, com taxa de verdadeiro positivo igual a 1 e taxa de falso positivo igual a 0. A curva ROC é construída ao variar o limite de decisão de um classificador, como as probabilidades posteriores de um classificador ingênuo de Bayes. A Área Sob a Curva (AUC) é calculada com base nessa curva ROC e serve para caracterizar o desempenho de um modelo de classificação.

Ao considerar todos os possíveis valores do limite de corte c, a curva ROC pode ser construída como um gráfico de sensibilidade (TPR) versus 1-especificidade (FPR). Para qualquer limite de corte c, podemos definir (Cali; Longobardi, 2015):

Figura 29 – Equação dos valores Sensibilidade e Especificidade.

$$\mathrm{TPR}(c) = \mathcal{P}(T \geq c|E+)$$
$$\mathrm{FPR}(c) = \mathcal{P}(T \geq c|E-).$$

Fonte: Cali e Longobardi - Mathematical properties of the ROC curve and their applications.

Dessa forma a curva ROC é representada por:

Figura 30 – Equação da curva ROC.

$$\mathrm{ROC}(\cdot) = \{\mathrm{FPR}(c), \mathrm{TPR}(c), \quad c \in (-\infty, +\infty)\}$$

Fonte: Cali e Longobardi - Mathematical properties of the ROC curve and their applications.

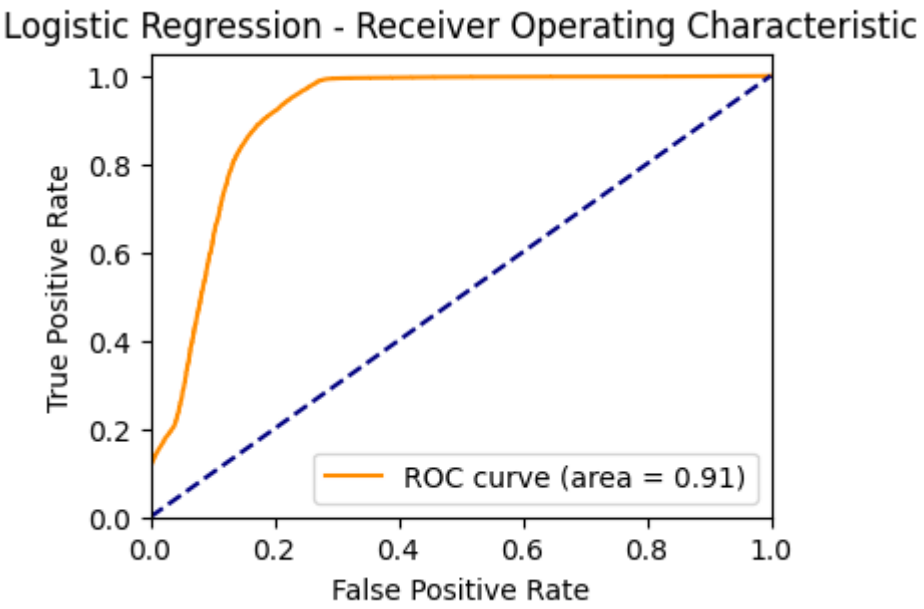
Onde a função ROC mapeia t para $TPR(c)$ e c é o corte correspondente a $FPR(c) = t$.

A curva inclinada para cima é um indicativo positivo para o modelo de classificação. Isso significa que o modelo é capaz de alcançar altas taxas de verdadeiro positivo com baixa taxa de falsos positivos. A área = 0.94 sugere que o modelo tem uma boa capacidade de distinguir entre as classes positivas e negativas, sendo que um valor próximo de 1 indica um melhor desempenho do modelo.

Através da curva ROC foi possível fazer comparações em relação aos outros algoritmos observado nas figuras 31 a 33:

- Logistic Regression:

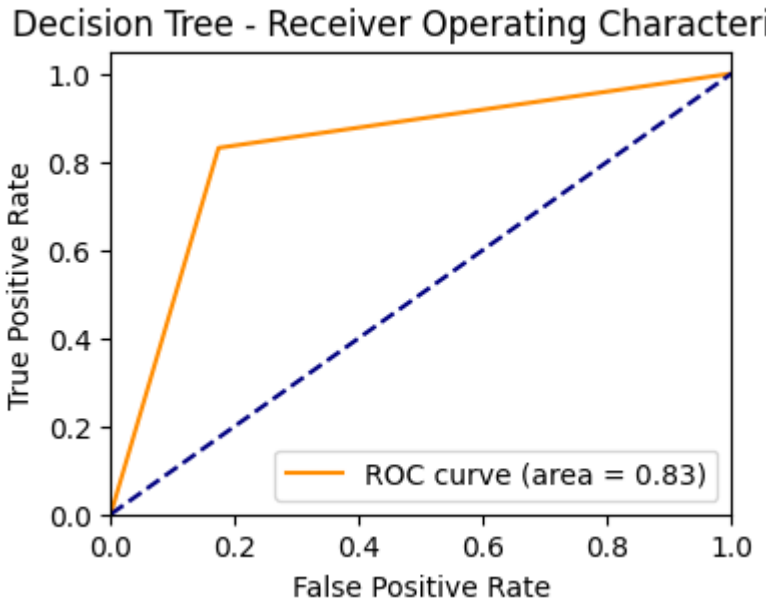
Figura 31 – Gráfico ROC – Logistic Regression.



Fonte: Elaboração Própria.

- Decision Tree:

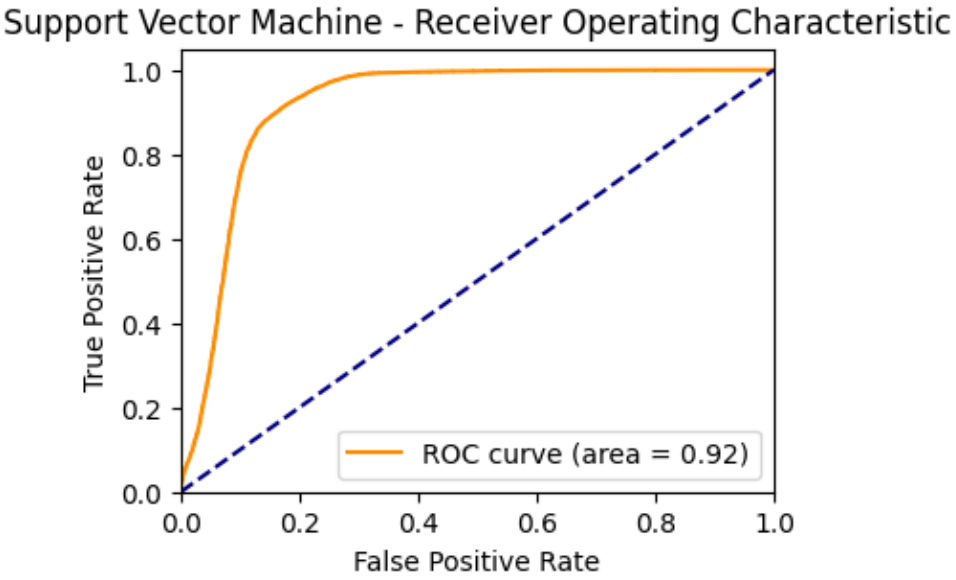
Figura 32 – Gráfico ROC – Decision Tree.



Fonte: Elaboração Própria.

- Support Vector Machine (SVM):

Figura 33 – Gráfico ROC – Support Vector Machine.



Fonte: Elaboração Própria.

Os gráficos mostram que estes algoritmos conseguiram ter uma boa capacidade de distinguir entre as classes positivas e negativas, mas ainda assim tendo uma eficiência menor que XGBoost.

Observando os algoritmos de *Gradient Boosting* e *Convoluncional Neural Networ*k (CNN) figura 34 e 35:

Figura 34 – Gradient Boosting, relatório de classificação e métricas.

XGBoost Classification Report				
	Precision	***Recall***	***f1-score***	***Support***
1 (*True*)	0.92	0.81	0.86	16239
0 (*False*)	0.84	0.93	0.88	17337
Accuracy			0.87	33576
Macro avg	0.88	0.87	0.87	33576
Weighted avg	0.88	0.87	0.87	33576

Fonte: Elaboração Própria.

Figura 35 – CNN, relatório de classificação e métricas.

XGBoost Classification Report				
	Precision	***Recall***	***f1-score***	***Support***
1 (*True*)	0.93	0.79	0.85	16239
0 (*False*)	0.83	0.94	0.88	17337
Accuracy			0.87	33576
Macro avg	0.88	0.87	0.87	33576
Weighted avg	0.88	0.87	0.87	33576

Fonte: Elaboração Própria.

Os dois algoritmos obtiveram valores de 0.94 nos resultados do gráfico ROC, o mesmo valor alcançado pelo XGBoost, mas tiveram muitas variações entre *precision* e *recall* em suas métricas, apesar de terem números maiores em relação a alguns pontos comparado ao XGBoost, tanto o Gradient Boosting quanto o CNN, apresentaram oscilações nos treinamentos realizados, trazendo valores das curvas ROC às vezes menores com valores de 0.93, enquanto o XGBoost manteve constância em seus resultados, além do balanceamento de suas métricas, tendo poucas variações, mostrando uma melhor eficiência, tanto para interpretação de dados, quanto na velocidade de processamento dos mesmos.

9 CONCLUSÃO

No presente artigo, foi descrita uma solução para o Spotify Sequential Skip Prediction Challenge, por meio de análises exploratórias, análise dos dados fornecidos e aplicação de teorias para o entendimento do uso dos modelos e algoritmos necessários.

A abordagem consistiu na identificação das melhores características da base de dados fornecida, limpeza do Data Frame de dados considerados pouco relevantes e criação de um pipeline para gerar previsões usando vários modelos de classificação diferentes, treinados com os recursos disponibilizados de cada faixa e características comportamentais dos usuários da plataforma Spotify.

Uma das vantagens da solução proposta é a capacidade de implementar novos modelos dentro do pipeline criado, bem como a possibilidade de adicionar mais recursos aos modelos, caso necessário. O uso do modelo de Random Forest e SHAP Values para classificação e identificação das características mais importantes (Feature importances) auxiliou no entendimento de quais recursos podem gerar melhores resultados.

Com essa abordagem, o problema de pesquisa foi respondido e os objetivos foram alcançados, obtendo-se uma acurácia de 87% nas previsões de pulos de faixas dos usuários.

Entretanto, a solução apresentou algumas limitações, como o poder de processamento limitado, que poderia ter proporcionado resultados mais precisos com conjuntos de dados maiores. Além disso, devido à restrição de tempo, não foi possível avaliar todas as combinações relacionadas às características acústicas. Outra dificuldade foi a limitação dos dados fornecidos pelo Spotify, uma vez que dados de alto nível dos recursos musicais, bem como características mais abrangentes dos artistas e comportamentos dos usuários, poderiam melhorar os resultados.

Como recomendação para estudos futuros, sugere-se o uso de modelos de deep learning mais robustos, juntamente com o uso dos dados completos fornecidos, a fim de obter resultados mais interessantes. No entanto, é importante considerar que essa abordagem exigirá um poder de processamento e recursos computacionais substanciais. Além disso, alinhar as predições de pulos de faixas com modelos de recomendação também pode ser um caminho a ser explorado, visto que a predição das faixas pode contribuir diretamente nos processos de recomendação do aplicativo.

SPOTIFY SEQUENTIAL SKIP PREDICTION CHALLENGE: Exploration and construction of prediction models of skipping behavior of musical tracks.

ABSTRACT

Spotify is one of the biggest music streaming platforms today. With the growing number of users and artists, a high volume of data is generated, which makes its exploration essential. Recommender systems play an important role in these services, mainly by providing personalized ‘Playlists’ for their users. Exploring and understanding clients' interactions with sessions can be beneficial to understanding their preferences in the context of each session. Thus, the company created the Spotify Senquential Skip Prediction Challenge, an open challenge focused on predicting whether a music track in a session will be skipped by the user. The objective of this work was to explore the data provided and build a model that could predict whether a platform user would skip a track, aiming to better understand their interactions with the platform itself. Through exploration and mining of the database to obtain the best features, a pipeline with eight machine learning algorithms was built to predict each user's skips. In this way, it was possible to observe the capacity that the XGBoost algorithm has and its efficiency in solving data classification problems, presenting good accuracy in solving the challenge.

Key words: Machine Learning; track skip; music; prediction models; music recommendation system.

REFERÊNCIAS

BHATTACHARYA, Sweta et al. A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU. **Electronics**, v. 9, n. 2, p. 219, 2020.

BROST, B.; MEHROTRA, R.; JEHAN, T. The music streaming sessions dataset. **In Proceedings of the 2019 Web Conference**. ACM, 2019.

CALÌ, Camilla; LONGOBARDI, Maria. Some mathematical properties of the ROC curve and their applications. **Ricerche di Matematica**, v. 64, p. 391-402, 2015.

CELMA, O.; HERRERA, P. New Challenges in Music Recommender Systems Research. **In Proceedings of the ACM RecSys Workshop on Music Recommendation and Discovery (MUSIC)**, 2008.

CHANG, Sungkyun; LEE, Seungjin; LEE, Kyogu. Sequential skip prediction with few-shot in streamed music contents. **arXiv preprint arXiv:1901.08203**, 2019. URL http://arxiv.org/abs/1901.08203.

CHEN, C., Li, C.; OGIHARA, M. . Understanding User Behavior in Online Music-Sharing Social Networks. **Journal of the American Society for Information Science and Technology (JASIST)**, v. 63, n. 2, p. 429-445, 2012.

CHEN, T; Guestrin, C. XG Boost. In: **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. San Francisco, CA, USA, 2016. p. 785-794.

GRANDINI, Margherita; BAGLI, Enrico; VISANI, Giorgio. Metrics for multi-class classification: an overview. **arXiv preprint arXiv:2008.05756**, 2020.

HURTADO, Alex; WAGNER, Markie; MUNDADA, Surabhi. Thank you, Next: Using NLP Techniques to Predict Song Skips on Spotify based on Sequential User and Acoustic Data, 2019.

KATZ, M. L. The Economics of Network Industries. **Springer Science & Business Media**, 2010.

LIEW, Xin Yu; HAMEED, Nazia; CLOS, Jeremie. An investigation of XGBoost-based algorithm for breast cancer classification. **Machine Learning with Applications**, v. 6, p. 100154, 2021.

LIU, Yuchun et al. Diagnosis of Parkinson's disease based on SHAP value feature selection. **Biocybernetics and Biomedical Engineering**, v. 42, n. 3, p. 856-869, 2022.

LOUDLAB. **Why is my track getting skipped on Spotify?**. Disponível em: https://www.loudlab.org/blog/why-is-my-track-getting-skipped-on-spotify/#. Acesso em: 30 de maio de 2023.

LUNDEN, Ingrid. **Spotify hits 155M paid subscribers and 345M total active users**. Disponível em: https://techcrunch.com/2021/11/03/spotify-hits-155m-paid-subscribers-and-345m-total-active-users/. Acesso em: 8 de junho de 2023.

MAMEDE, **Mario. A História do Spotify**. Disponível em: https://www.showmetech.com.br/historia-do-spotify/. Acesso em: 30 maio 2023.

MEGGETTO, Francesco et al. On skipping behaviour types in music streaming sessions. In: **Proceedings of the 30th ACM International Conference on Information & Knowledge Management**. 2021.

MOFFITT, C. **Overview of Pandas Data Types. 2018**. Disponível em: https://pbpython.com/pandas_dtypes.html. Acesso em: 16 jun. 2023.

OLIVEIRA, B. **"Coeficientes de Correlação."** Agosto 23, 2019. Disponível em: https://statplace.com.br/blog/coeficientes-de-correlacao/. Acesso em: 16 jun. 2023.

POYAR, Kyle. **Tudo que você precisa saber sobre o modelo Freemium em 2020**. Disponível em: https://hack.consulting/tudo-que-voce-precisa-saber-sobre-o-modelo-freemium-em-2020/#/. Acesso em: 19 de junho de 2023.

RASCHKA, Sebastian. **An Overview of General Performance Metrics of Binary Classifier Systems**. E-mail: se.raschka@gmail.com. Outubro 21, 2014.

RONAGHAN, Stacey. The Mathematics of Decision Trees, Random Forest and Feature Importance in Scikit-learn and Spark. **Towards Data Science**, [S.l.], v. 6, n. maio, p. 11, 2018. Disponível em: https://towardsdatascience.com/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3. Acesso em: 13 jun. 2023.

SAINI, **A. An Introduction to Random Forest Algorithm for beginners**. 2021. Disponível em: https://www.analyticsvidhya.com/blog/2021/10/an-introduction-to-random-forest-algorithm-for-beginners/. Acesso em: 26 ago. 2022.

SCHEDL, M.; HAUGER, D. The LFM-1b Dataset for Music Retrieval and Recommendation. **In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)**, 2015.

SCHEDL, M.; FLEXER, A. Improved Estimations of Skip Bias in Music Recommendation. **In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)**, 2016.

SHAIKH, Rahil. **Choosing the right Encoding Method-Label vs OneHot Encoder**. 2018. Disponível em: https://towardsdatascience.com/choosing-the-right-encoding-method-label-vs-onehot-encoder-a4434493149b. Acesso em 12 jun. 2023.

SILVA, R. O.; SILVA, I. R. S. Linguagem de Programação Python. **Tecnologias em Projeção**, v. 10, n. 1, p. 55-71, 2019.

Significado de Correlação. Disponível em: https://www.significados.com.br/correlacao/. Acesso em: 9 de junho de 2023.