MODELOS DE ÁRVORE SUPERAM TRANSFORMERS NA PREVISÃO DE ENGAJAMENTO DE MUSICAS

João Victor de Souza Albuquerque

RESUMO

YouTube e Spotify revolucionaram o consumo de mídia ao permitirem que criadores de conteúdo alcancem audiências globais sem intermediários tradicionais. A predominância de músicas no engajamento dos usuários impulsiona o desenvolvimento de modelos preditivos para estimar sua popularidade. Este trabalho investiga a eficácia do modelo *FT-Transformer*, voltado para dados tabulares, em comparação com métodos tradicionais baseados em árvores, como Random Forest e XGBoost. Utilizando um conjunto de dados real com metadados de músicas e vídeos do Spotify e do YouTube, os resultados indicam que os modelos baseados em árvores continuam superiores na modelagem de dados tabulares heterogêneos e de baixa dimensionalidade. Embora o *FT-Transformer* não tenha superado esses modelos, apresentou desempenho competitivo em relação ao XGBoost, especialmente com suporte a GPU. Esses achados reforçam as limitações dos Transformers nesse contexto e apontam caminhos para pesquisas futuras que integrem dados tabulares com informações multimodais, como letras de músicas ou elementos visuais dos vídeos.

Palavras-chave: FT-Transformer; Dados Tabulares; Predição de Engajamento Musical; Random Forest; XGBoost

ABSTRACT

YouTube and Spotify have revolutionized media consumption by allowing content creators to reach global audiences without traditional intermediaries. The prominence of music in user engagement drives the development of predictive models to estimate its popularity. This study investigates the effectiveness of the *FT-Transformer* model, designed for tabular data, in comparison with traditional tree-based methods such as Random Forest and XGBoost. Using a

real dataset composed of metadata from Spotify and YouTube, the results show that tree-based models remain superior in modeling low-dimensional and heterogeneous tabular data. Although the *FT-Transformer* did not outperform the traditional methods in any of the evaluated scenarios, it showed competitive performance against XGBoost, especially in GPU-supported environments. These findings reinforce existing literature on the limitations of Transformers in tabular data contexts and suggest promising directions for future research that combines tabular data with multimodal sources, such as song lyrics or visual content from videos.

Keywords: FT-Transformer; Tabular Data; Music Engagement Prediction; Random Forest; XGBoost;

1 INTRODUÇÃO

As plataformas digitais, como o YouTube e o Spotify, revolucionaram a distribuição de conteúdo ao democratizar a criação e permitir o acesso global, sem a necessidade de intermediários tradicionais. A música, em particular, destaca-se como o tipo de conteúdo mais consumido PEDROSO *et al.* (2016), o que impulsiona o desenvolvimento de ferramentas preditivas voltadas à estimativa de popularidade ou engajamento. Segundo dados da UBC UBC (2025), o mercado musical brasileiro faturou R$3,486 bilhões em 2024, enquanto a receita global da indústria fonográfica ultrapassou US$29 bilhões, evidenciando a relevância econômica do setor e o papel estratégico de modelos preditivos confiáveis.

Nos últimos anos, a arquitetura Transformer VASWANI e al. (2017) tornou-se referência em tarefas de aprendizado profundo, sendo amplamente adotada em áreas como processamento de linguagem natural. Seu sucesso motivou adaptações para domínios distintos, incluindo dados tabulares. Entre essas propostas, destaca-se o FT-Transformer GORISHNIY *et al.* (2021), projetado para lidar com variáveis numéricas e categóricas por meio de codificações aprendíveis.

Entretanto, modelos baseados em árvores, como o XGBoost e o Random Forest, ainda são considerados padrão em aplicações com dados tabulares, especialmente pela robustez e desempenho observados em diversos contextos práticos GRINSZTAJN *et al.* (2022). Estudos recentes indicam que redes neurais, mesmo profundas, podem ter dificuldades em lidar com irregularidades e atributos não informativos presentes nesse tipo de dado.

Além disso, a literatura sobre predição de popularidade musical em plataformas digitais tem se concentrado majoritariamente em métodos tradicionais, sem explorar a fundo abordagens

modernas baseadas em Transformers PAREEK *et al.* (2022), Yee e Raheem (2022). Assim, permanece em aberto a questão sobre a viabilidade de aplicar essas arquiteturas com vantagem em contextos tabulares reais e heterogêneos.

Neste trabalho, buscamos responder a essa questão avaliando o desempenho de modelos modernos, como o FT-Transformer, em comparação com modelos tradicionais em uma tarefa de regressão que visa prever o engajamento de músicas no YouTube. Utilizamos o conjunto de dados "Spotify and YouTube" Rastelli *et al.* (2022), complementado por atributos temporais. Os modelos comparados incluem Regressão Linear, Random Forest, XGBoost e Perceptron Multicamadas (MLP), avaliados tanto com hiperparâmetros padrão quanto otimizados via *grid search* e Optuna AKIBA *et al.* (2019).

Os resultados obtidos estão alinhados com as conclusões de GRINSZTAJN *et al.* (2022). Mais especificamente, os modelos baseados em árvores apresentaram o melhor desempenho em todos os cenários analisados. Os resultados são discutidos à luz da literatura, demonstrando como este estudo contribui para o avanço das investigações recentes sobre a aplicação de Transformers em tarefas de predição com dados tabulares.

A organização deste trabalho é a seguinte: o Capítulo 2 apresenta os trabalhos relacionados; o Capítulo 3 descreve o conjunto de dados e fundamentos teóricos; o Capítulo 4 detalha a metodologia adotada; os resultados e discussões são apresentados no Capítulo 5; e as conclusões encontram-se no Capítulo 6.

2 TRABALHOS RELACIONADOS

A maioria dos estudos relacionados à predição de desempenho musical em plataformas de streaming concentra-se no Spotify. Muitos desses trabalhos aplicam diferentes modelos de aprendizado de máquina e propõem diversas técnicas de pré-processamento, como observado nos estudos a seguir.

PAREEK *et al.* (2022) investigaram a previsão da popularidade de músicas utilizando as características de áudio fornecidas pelo Spotify, tais como intensidade sonora (*loudness*), energia e acústicidade (*acousticness*). Os autores aplicaram três classificadores de aprendizado de máquina — Random Forest, K-Nearest Neighbors (KNN) e Linear Support Vector Classifier (LSVC) — com o objetivo de classificar músicas como populares ou não. Os resultados indicaram que o classificador Random Forest superou os demais, obtendo os melhores valores de acurácia, precisão, revocação e F1-score, demonstrando, assim, sua eficácia na predição da popularidade

musical com base exclusivamente em atributos de áudio.

Yee e Raheem (2022) avaliaram se a combinação de características de áudio do Spotify com métricas de redes sociais extraídas do YouTube poderia melhorar a previsão da popularidade de uma música. O estudo construiu um conjunto de dados com faixas recém-lançadas, reunindo atributos de áudio do Spotify e dados sociais (como visualizações, curtidas e comentários) do YouTube. A popularidade foi quantificada a partir de cinco métricas derivadas da parada *Top 200 Daily* do Spotify. Modelos de aprendizado de máquina foram treinados utilizando dois conjuntos de atributos: um contendo apenas as características de áudio e outro combinando áudio com dados de redes sociais. Os resultados mostraram que a inclusão de dados sociais aumentou significativamente o desempenho dos modelos, com melhorias de acurácia entre 10% e 60%, evidenciando o valor de dados multimodais na tarefa de previsão de popularidade musical.

Outros trabalhos exploraram técnicas mais avançadas de aprendizado profundo para desenvolver modelos especializados na predição da popularidade de músicas no Spotify. Essas abordagens frequentemente extrapolam os métodos tradicionais ao proporem estratégias de pré-processamento personalizadas para lidar com as particularidades do problema. Por exemplo, MARTÍN-GUTIERREZ *et al.* (2020) introduziram o HitMusicNet, uma arquitetura inovadora de aprendizado profundo *end-to-end* projetada para prever a popularidade de gravações musicais. O modelo integra diferentes modalidades de dados, incluindo atributos de áudio, metadados e, potencialmente, outras informações relevantes, a fim de capturar os diversos fatores que influenciam o sucesso de uma música. Os autores também apresentaram o conjunto de dados SpotGenTrack Popularity Dataset (SPD), com o objetivo de facilitar a comparação entre diferentes modelos preditivos nesse domínio. O HitMusicNet demonstrou bom desempenho na tarefa de previsão de popularidade musical, destacando a eficácia de abordagens multimodais com aprendizado profundo.

Este estudo distingue-se dos trabalhos anteriores por concentrar-se na análise da capacidade preditiva de diferentes modelos aplicados especificamente à previsão de engajamento musical na plataforma YouTube. Além disso, realiza uma comparação entre modelos tradicionais — amplamente utilizados em tarefas com dados tabulares — e uma arquitetura moderna (FT-Transformer), projetada especificamente para esse tipo de dado.

3 FUNDAMENTAÇÃO TEÓRICA

3.1 Descrição do Conjunto de Dados

Esta seção apresenta o conjunto de dados utilizado para o treinamento e avaliação dos modelos preditivos. Os dados foram obtidos a partir do repositório "Spotify and YouTube" Rastelli *et al.* (2022), o qual reúne informações estatísticas e descritivas de dez músicas distintas, provenientes de diferentes artistas.

O conjunto contém aproximadamente 20 mil amostras e 26 atributos, os quais são organizados em duas categorias: características musicais e características descritivas.

As características musicais representam propriedades técnicas das faixas, geralmente extraídas automaticamente por ferramentas de análise de áudio. Entre elas, destacam-se:

- **Danceability**: Mede a adequação da faixa para dança, variando entre 0,0 (menos dançante) e 1,0 (mais dançante).
- **Energy**: Refere-se à percepção de intensidade e atividade da faixa, em escala de 0,0 a 1,0.
- **Key**: Tonalidade da música, expressa como valor inteiro com base na notação de classe de alturas (*Pitch Class*).
- **Loudness**: Intensidade sonora média da faixa, medida em decibéis (dB).
- **Speechiness**: Estima a presença de fala na faixa. Valores mais altos (próximos de 1,0) indicam conteúdo predominantemente falado.
- **Acousticness**: Probabilidade de que a faixa seja acústica, em escala de 0,0 a 1,0.
- **Instrumentalness**: Indica a ausência de vocais na faixa.
- **Liveness**: Detecta a presença de público ao vivo. Valores mais elevados sugerem gravações realizadas em apresentações ao vivo.
- **Valence**: Reflete a positividade emocional da faixa, de 0,0 (negativa) a 1,0 (positiva).
- **Tempo**: Ritmo médio da música, em batidas por minuto (BPM).
- **Duration_ms**: Duração total da faixa em milissegundos.

As características descritivas incluem informações sobre os metadados e o desempenho da faixa nas plataformas Spotify e YouTube. São elas:

- **Track**: Nome da faixa no Spotify.
- **Artist**: Nome(s) do(s) artista(s) responsáveis pela faixa.
- **Album**: Título do álbum ao qual a música pertence.
- **Album_type**: Indica se a música foi lançada como parte de um álbum ou como single.
- **Uri**: Identificador utilizado pela API do Spotify para localizar a faixa.
- **Url_youtube**: Endereço do vídeo correspondente no YouTube.
- **Title**: Título do vídeo publicado na plataforma.
- **Channel**: Nome do canal do YouTube responsável pela publicação.
- **Description**: Texto descritivo do vídeo.
- **Licensed**: Indica se o vídeo contém material licenciado.
- **Official_video**: Valor booleano que indica se o vídeo é o clipe oficial.
- **Stream**: Número de reproduções da faixa no Spotify.
- **Views**: Número de visualizações do vídeo no YouTube.
- **Likes**: Total de curtidas atribuídas ao vídeo.
- **Comments**: Número de comentários recebidos no vídeo.

Adicionalmente, foram incluídas duas variáveis temporais: a data de lançamento do vídeo no YouTube e a data de publicação da faixa no Spotify. Essa inclusão deve-se à constatação de que alguns atributos variam significativamente ao longo do tempo.

3.2 Modelos

Esta seção apresenta uma visão geral dos modelos utilizados neste trabalho.

3.2.1 Random Forest

O Random Forest é um método de aprendizado por conjunto (*ensemble*) que constrói múltiplas árvores de decisão durante o treinamento e retorna a média das predições (em tarefas de regressão) das árvores individuais. Ele introduz aleatoriedade na seleção de subconjuntos de dados e atributos para cada árvore, o que contribui para a redução do sobreajuste e aumenta sua capacidade de generalização. É capaz de capturar relações não lineares complexas entre os atributos sem exigir pré-processamento extensivo. O Random Forest é amplamente utilizado em domínios científicos e industriais devido à sua robustez e alta acurácia preditiva BREIMAN (2001), LOUPPE (2014).

3.2.2 Extreme Gradient Boosting (XGBoost)

De acordo com JIN e al. (2019), o Extreme Gradient Boosting (XGBoost) é uma implementação otimizada do método de *boosting*, cuja proposta é construir um preditor forte por meio da combinação de múltiplos classificadores fracos. Conforme FRIEDMAN (2001), citado por ZERAATGARI e al. (2023), o XGBoost é uma versão avançada do algoritmo de *boosting* por gradiente, reconhecida por sua alta eficiência, flexibilidade e aplicabilidade. JIN e al. (2019) destacam, por exemplo, aplicações do XGBoost em astronomia, como a separação de sinais de pulsares e a classificação de fontes não identificadas no catálogo Fermi-LAT.

3.2.3 Regressão Linear

A regressão linear é um dos modelos mais fundamentais e amplamente utilizados para tarefas de regressão. Assume-se uma relação linear entre as variáveis de entrada e a variável alvo, sendo os coeficientes estimados com o objetivo de minimizar o erro entre os valores previstos e os observados — geralmente utilizando o Erro Quadrático Médio (MSE) como função de perda. Apesar de sua simplicidade, a regressão linear é frequentemente empregada como modelo de base e apresenta desempenho satisfatório quando a relação entre os atributos e a saída é aproximadamente linear. Sua interpretabilidade e baixo custo computacional tornam-na uma ferramenta valiosa em contextos de pesquisa e aplicação prática JAMES *et al.* (2013).

3.2.4 Perceptron Multicamadas (MLP)

O Perceptron Multicamadas (MLP) é uma classe de redes neurais artificiais composta por uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Cada camada oculta aplica uma transformação linear seguida por uma função de ativação não linear à entrada recebida. O processo de aprendizado ocorre por meio do algoritmo de retropropagação, que ajusta os pesos da rede com base na minimização de uma função de perda previamente definida ZERAATGARI e al. (2023). Redes MLP podem conter múltiplas camadas ocultas e diversos neurônios, o que permite a modelagem de padrões complexos nos dados.

3.2.5 FT-Transformer

A arquitetura Transformer, proposta por VASWANI e al. (2017), representou um marco no desenvolvimento de modelos de aprendizado profundo para tarefas de modelagem sequencial, ao substituir estruturas recorrentes e convolucionais por um mecanismo inteiramente baseado em atenção. Seu principal componente é o mecanismo de autoatenção com múltiplas cabeças (*multi-head*), que possibilita a captura de dependências de longo alcance entre os elementos da sequência. A arquitetura é composta por blocos empilhados de codificadores e decodificadores, contendo camadas de autoatenção, redes neurais *feedforward*, normalização de camadas e conexões residuais. Além disso, codificações posicionais são adicionadas aos *embeddings* de entrada para preservar a ordem sequencial. Essa estrutura altamente paralelizável contribuiu para a adoção do Transformer como base de diversos modelos de ponta em Processamento de Linguagem Natural (PLN).

Neste trabalho, utilizou-se uma versão adaptada do Transformer para dados tabulares, denominada FT-Transformer, introduzida por GORISHNIY *et al.* (2021). Diferentemente da arquitetura original, projetada para entradas sequenciais, o FT-Transformer opera sobre conjuntos de atributos tabulares que incluem variáveis numéricas e categóricas. Cada atributo é codificado como um *token* por meio de *embeddings* aprendíveis, permitindo que o mecanismo de atenção capture relações entre atributos de maneira orientada aos dados. A arquitetura emprega um *token* [CLS] para agregação das informações, de modo similar ao BERT, e utiliza um *Feature Tokenizer* para lidar com entradas heterogêneas. Essa abordagem tem demonstrado desempenho competitivo em diversos benchmarks com dados estruturados, oferecendo uma alternativa promissora aos modelos tradicionais, como algoritmos de *boosting* e redes neurais densamente conectadas.

4 METODOLOGIA

4.1 Pré-processamento dos Dados

4.1.1 Engenharia de Dados

Além das colunas já existentes, foram criadas novas variáveis para enriquecer o conjunto de dados. A primeira, *days_on_platform*, representa o número de dias desde a publicação de um vídeo na plataforma. A segunda, *artist_number*, corresponde ao número total de artistas distintos envolvidos em determinada faixa.

A variável categórica *album_type* foi codificada por meio de *one-hot encoding* para os modelos tradicionais — Random Forest, XGBoost, Regressão Linear e Perceptron Multicamadas (MLP, do inglês *Multi-Layer Perceptron*). No caso do FT-Transformer, essa variável foi mantida em seu formato original, uma vez que o modelo é capaz de lidar diretamente com dados categóricos, sem necessidade de codificação adicional.

A variável *engagement_rate* foi construída para atuar como variável-alvo na tarefa de regressão. Essa métrica agrega indicadores-chave de desempenho — como curtidas, comentários e visualizações — em um único valor, oferecendo uma medida abrangente do engajamento do público. A fórmula utilizada foi:

$$\text{engajamento} = \frac{\text{curtidas} + \text{comentários}}{\text{visualizações}} \tag{4.1}$$

4.1.2 Seleção de Atributos

Após a etapa de engenharia de dados, conduziu-se um processo de seleção para definir quais variáveis seriam mantidas no conjunto final. Todas as variáveis musicais foram preservadas, de modo a permitir que os modelos aprendessem a prever o engajamento com base nas características sonoras da música, além de algumas variáveis descritivas.

As variáveis descritivas selecionadas foram: *artist_number*, *album_type*, *stream* e *engagement_rate* — sendo esta última a variável-alvo. As demais variáveis foram descartadas por apresentarem redundância ou por dificultarem a interpretação dos modelos. A variável *stream* foi mantida por ter demonstrado, em testes empíricos, melhoria no desempenho preditivo.

4.1.3 Limpeza dos Dados

Na etapa de limpeza, inicialmente eliminaram-se todas as linhas com valores ausentes nas variáveis selecionadas. Essa abordagem foi viável devido à grande quantidade de amostras disponíveis, tornando a perda de dados estatisticamente insignificante.

Em seguida, tratou-se a presença de outliers na variável-alvo. Para isso, os dados foram segmentados em faixas temporais: 0 a 30 dias, 31 a 90 dias, 91 a 365 dias e acima de 366 dias. Essa segmentação foi motivada pela observação de que o engajamento tende a diminuir com o tempo. Sem essa divisão, a aplicação de um limiar global para remoção de outliers poderia levar à exclusão indevida de vídeos recentes com alto engajamento.

Dentro de cada faixa, aplicou-se o método do Intervalo Interquartil (IQR), técnica estatística amplamente utilizada para detecção de valores anômalos TUKEY (1977). Os valores acima do limite superior foram removidos.

4.1.4 Divisão dos Dados

Com a base segmentada, selecionou-se a faixa de vídeos com até 30 dias na plataforma como amostra principal para divisão dos dados. Esta foi separada em dois subconjuntos: treino/validação (75%) e teste (25%).

Posteriormente, o subconjunto de treino/validação foi novamente dividido em treino (80%) e validação (20%). Para garantir reprodutibilidade, utilizaram-se sementes aleatórias nas divisões.

Buscando enriquecer os dados de treinamento, o subconjunto de faixas com até 30 dias foi combinado com os dados remanescentes (faixas com mais de 30 dias). Essa estratégia se justifica por dois motivos: (i) permitir a avaliação do modelo na previsão de engajamento de músicas recém-lançadas, e (ii) observar que a inclusão de faixas antigas resultou em ganhos expressivos de desempenho. Assim, os dados antigos foram utilizados exclusivamente no treinamento.

4.1.5 Normalização

A normalização foi aplicada de maneira diferenciada entre os modelos tradicionais e o FT-Transformer.

Para os modelos tradicionais, utilizou-se a normalização Min-Max, implementada pela função `MinMaxScaler()` da biblioteca `scikit-learn`, aplicada às colunas preditoras X.

No caso do FT-Transformer, seguiu-se a implementação oficial dos autores disponível

no GitHub GORISHNIY *et al.* (2021). Os atributos X foram normalizados com `Quantile Transformer()`, em conjunto com injeção de ruído gaussiano, conforme descrito na configuração experimental original. A variável-alvo y foi normalizada com base no desvio padrão, segundo a função de normalização definida no repositório oficial.

A Figura 1a apresenta o pipeline completo de pré-processamento.

4.2 Seleção de Modelos

Foram considerados dois cenários experimentais: (1) sem otimização de hiperparâmetros, e (2) com otimização utilizando GridSearch ou Optuna.

No primeiro cenário, avaliou-se o desempenho dos modelos com configurações padrão fornecidas pela biblioteca `scikit-learn`, simulando o uso por usuários não especializados. Cada modelo foi executado em dez rodadas independentes, utilizando os dados previamente divididos, sem separação adicional entre treino e validação.

No segundo cenário, investigou-se o impacto da seleção de hiperparâmetros na performance dos modelos. Para cada modelo, realizaram-se dez rodadas independentes, coletando a média e o desvio padrão do MSE, bem como a melhor configuração individual.

As técnicas empregadas para a seleção foram GridSearch (apenas para os modelos clássicos, devido ao custo computacional) e Optuna AKIBA *et al.* (2019), que aplica otimização bayesiana.

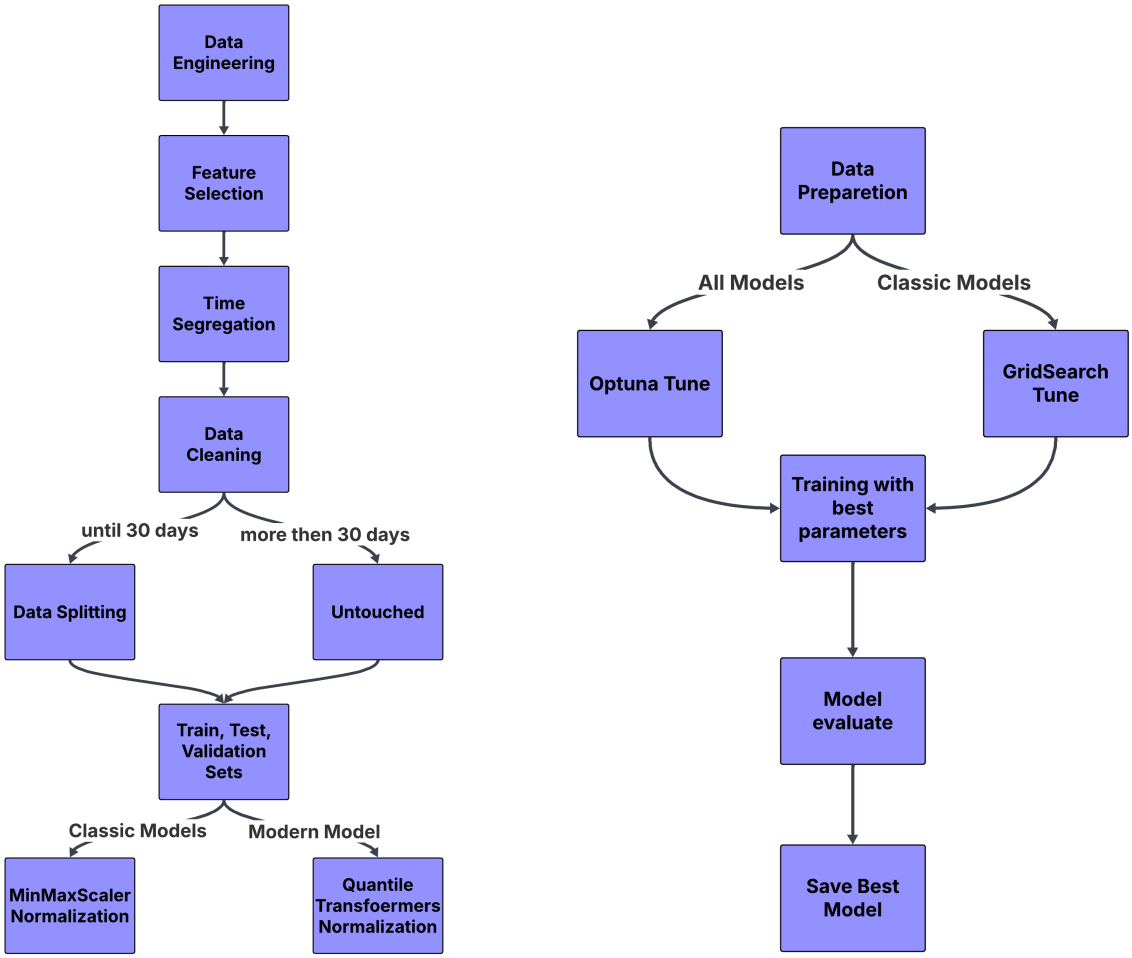
A cada rodada, os dados foram divididos e normalizados conforme as etapas anteriores. O modelo foi treinado com o conjunto de treino/validação e avaliado sobre o conjunto de teste. A configuração com menor MSE foi retida para análise comparativa.

A Figura 1b apresenta o pipeline completo da seleção de modelo.

4.3 Avaliação

Conforme discutido anteriormente, utilizou-se o Erro Quadrático Médio (MSE) como métrica principal de avaliação. Os valores obtidos ao longo das dez rodadas foram utilizados para calcular a média e o desvio padrão do desempenho de cada modelo.

A avaliação foi conduzida em três etapas: (i) comparação entre os modelos com hiperparâmetros padrão; (ii) análise do ganho obtido após otimização de hiperparâmetros em relação à versão padrão de cada modelo; e (iii) comparação entre os melhores modelos ajustados para identificar aquele com melhor desempenho geral.



(a) Fluxo de pré-processamento. (b) Fluxo de seleção de modelo

Figura 1 – Fluxo de preprocessamento e seleção de modelos. (a) Fluxo de pré-processamento: coleta, limpeza, seleção e normalização dos dados. (b) Fluxo de seleção de modelos: divisão em treino/validação/teste, otimização de hiperparâmetros (GridSearch e Optuna) e avaliação final via MSE.

Fonte: Elaborado pelo autor.

5 RESULTADOS

5.1 Desempenho dos Modelos com Parâmetros Padrão

Ao analisar os resultados obtidos pelos modelos com configurações padrão (Tabela 1), observa-se que o Regressor Linear apresentou desempenho significativamente inferior aos demais. Essa limitação está relacionada à sua incapacidade de modelar relações não lineares complexas nos dados, o que compromete seu poder preditivo. Ademais, por não possuir hiperparâmetros ajustáveis, esse modelo apresenta menor flexibilidade para melhorias de desempenho.

Em contrapartida, os modelos XGBoost, FT-Transformer e Random Forest exibiram resultados semelhantes, o que dificulta a identificação de um modelo mais adequado para esta

tarefa de regressão, nesta etapa. Embora o modelo MLP não tenha superado os demais em sua configuração padrão, seu desempenho foi relativamente próximo ao dos melhores modelos, indicando seu potencial como alternativa viável após ajustes de parâmetros.

Destaca-se, ainda, o desempenho do FT-Transformer, que, mesmo com parâmetros padrão, superou o MLP — uma arquitetura frequentemente aplicada em tarefas com dados tabulares — e apresentou desempenho comparável ao XGBoost, amplamente reconhecido por sua eficiência e capacidade preditiva. O Random Forest foi o modelo com melhor desempenho nesta configuração inicial.

Tabela 1 – Desempenho dos Modelos com Parâmetros Padrão em Dez Execuções Independentes

Modelo	***Média***	***Desvio Padrão***
Regressor Linear	7.2316	0.7596
Random Forest	3.1358	0.3833
MLP	4.1114	0.4997
XGBoost	3.3707	0.4148
FT-Transformer	3.5601	0.6433

Fonte: Elaborado pelo autor.

5.2 Desempenho dos Modelos Otimizados

Ao analisar o desempenho final do MLP na tarefa proposta, observa-se que a média do MSE obtida pelas estratégias de ajuste de hiperparâmetros com Optuna e GridSearch foi, na verdade, inferior à obtida com a versão padrão. Esse comportamento foi observado, em alguma medida, em todos os modelos, possivelmente devido a divisões dos dados que resultaram em conjuntos de treinamento com poucas amostras recentes, prejudicando a capacidade de generalização. Além disso, ao comparar o melhor resultado do MLP (identificado via Optuna) com as melhores versões dos demais modelos, o MLP permaneceu com o pior desempenho, apresentando um MSE aproximadamente 0,4 pontos superior.

Com relação ao desempenho do FT-Transformer em comparação ao XGBoost e ao Random Forest, observa-se que o MSE médio do FT-Transformer foi aproximadamente 0,47 pontos superior ao do Random Forest e 0,48 pontos superior ao do XGBoost. Esses resultados estão em consonância com as conclusões de GRINSZTAJN *et al.* (2022), que apontam que modelos baseados em árvores frequentemente superam métodos de aprendizado profundo em tarefas

Tabela 2 – Desempenho dos Modelos Otimizados com Duas Estratégias de Otimização

Modelo	***Otimizador***	***Média***	***Desvio Padrão***	***Melhor MSE***
Random Forest	GridSearch	3.3750	0.4742	2.5401
	Optuna	3.3223	0.4055	2.5819
MLP	GridSearch	4.9773	0.9203	3.5314
	Optuna	4.2563	0.5586	3.1479
XGBoost	GridSearch	3.3049	0.4324	2.7117
	Optuna	3.3086	0.4402	2.7636
FT-Transformer	Optuna	3.7982	0.9625	2.7647

Fonte: Elaborado pelo autor.

com dados tabulares. Contudo, ao considerar apenas a melhor execução do FT-Transformer, seu desempenho foi competitivo com o XGBoost — apresentando uma diferença de apenas 0,0011 no MSE. Dessa forma, a escolha entre os dois pode depender da disponibilidade de recursos computacionais: enquanto ambos obtiveram desempenhos semelhantes, o FT-Transformer é mais indicado em ambientes com suporte a GPU.

Ao comparar a melhor configuração do FT-Transformer com a melhor execução do modelo Random Forest, observa-se uma diferença mais expressiva, de 0,22 pontos no MSE. Esse resultado reforça a posição do Random Forest como o modelo mais eficaz para esta tarefa, tanto em termos de desempenho quanto pela sua simplicidade algorítmica e menor exigência computacional.

Com base nos estudos de GRINSZTAJN *et al.* (2022) e Charchyan (2024), constata-se que o FT-Transformer não apresentou o melhor desempenho na tarefa abordada. Um dos principais fatores reside na natureza dos dados de entrada e saída, que ainda apresentam variações abruptas — mesmo após o processo de normalização — conforme destacado por GRINSZTAJN *et al.* (2022). Essa limitação tornou-se ainda mais evidente durante experimentos preliminares, nos quais foi investigada a melhor forma de aplicação do FT-Transformer: a remoção da normalização do input (X) ou do target (y) resultou em quedas acentuadas de desempenho. Em contraste, como demonstrado por GRINSZTAJN *et al.* (2022), modelos baseados em árvores tendem a ser mais robustos a essas irregularidades.

Adicionalmente, a análise inicial dos dados levantou a hipótese de que o engajamento de uma música não é uniformemente influenciado por todas as suas características musicais. Algumas variáveis podem ter maior relevância, especialmente quando associadas a padrões recorrentes em determinados gêneros musicais — alguns dos quais são mais populares que outros. Essa hipótese

foi posteriormente corroborada por Charchyan (2024), que demonstraram que *danceability* e *loudness* são significativamente mais relevantes do que outras variáveis.

Embora esse desbalanceamento possa afetar todos os modelos, ele está diretamente relacionado à terceira hipótese de GRINSZTAJN *et al.* (2022), a qual aponta que a presença de variáveis com baixo sinal pode comprometer severamente o desempenho de redes neurais do tipo MLP. Esse foi um dos principais fatores indicados pelos autores para justificar a superioridade dos modelos baseados em árvores em contextos com dados tabulares. Ainda assim, esses achados indicam caminhos promissores para abordar o problema por meio de representações alternativas dos dados.

6 CONCLUSÃO

Considerando a tarefa de prever o engajamento de músicas na plataforma YouTube, a afirmação de GRINSZTAJN *et al.* (2022) — de que modelos baseados em árvores geralmente apresentam melhor desempenho em dados tabulares — mostrou-se válida no contexto deste estudo. Entre os modelos desse tipo, o Random Forest destacou-se como o que apresentou o melhor desempenho geral na tarefa proposta.

Por outro lado, o FT-Transformer demonstrou resultados competitivos em comparação ao XGBoost, sugerindo que, embora os modelos baseados em árvores tenham obtido melhor desempenho neste cenário específico, há indícios de que, em outras tarefas de predição com dados tabulares, o FT-Transformer possa alcançar resultados equivalentes ou superiores. Isso reforça a ideia de que as conclusões de GRINSZTAJN *et al.* (2022) indicam uma tendência robusta, embora não necessariamente generalizável a todos os contextos.

Adicionalmente, os resultados obtidos, em consonância com os achados de Charchyan (2024), apontam direções promissoras para investigações futuras. Tanto na predição de engajamento no YouTube quanto na previsão de popularidade no Spotify, uma linha de pesquisa relevante consiste na integração de atributos tabulares com informações textuais extraídas das letras das músicas, por meio de técnicas de Processamento de Linguagem Natural (PLN). Para tarefas específicas do YouTube, a combinação de dados tabulares com informações visuais provenientes dos vídeos, utilizando técnicas de processamento de imagem, representa uma abordagem promissora no aprimoramento dos modelos preditivos.

REFERÊNCIAS

AKIBA, T.; SANO, S.; YANASE, T.; OHTA, T.; KOYAMA, M. Optuna: A next-generation hyperparameter optimization framework. In: ACM. **Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining**. Anchorage, Alaska, EUA, 2019.

BREIMAN, L. Random forests. **Machine Learning**, Springer, Dordrecht, Holanda, v. 45, n. 1, p. 5–32, 2001.

CHARCHYAN, A. **Exploring Trends in Music Platforms: A Comparative Analysis of Key Factors for Trending Songs on Spotify and YouTube**. Tese (B.S. thesis) — American University of Armenia, Yerevan, Armenia, 2024. Disponível em: <https://tinyurl.com/mvb4p7tj>.

FRIEDMAN, J. H. Greedy function approximation: A gradient boosting machine. **Annals of Statistics**, Institute of Mathematical Statistics, v. 29, n. 5, p. 1189–1232, 2001.

GORISHNIY, Y.; RUBACHEV, I.; KHRULKOV, V.; BABENKO, A. Revisiting deep learning models for tabular data. In: **Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS)**. Sydney, Austrália: Curran Associates, Inc., 2021. v. 34, p. 18932–18943.

GRINSZTAJN, L.; OYALLON, E.; VAROQUAUX, G. Why do tree-based models still outperform deep learning on tabular data? **Advances in Neural Information Processing Systems**, v. 35, p. 22977–22990, 2022.

JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. **An Introduction to Statistical Learning: With Applications in R**. New York, EUA: Springer, 2013. (Springer Texts in Statistics).

JIN, X.; AL. et. A machine learning method to separate pulsar signals from noise. **Monthly Notices of the Royal Astronomical Society**, Oxford University Press, v. 485, n. 3, p. 3561–3574, 2019.

LOUPPE, G. **Understanding Random Forests: From Theory to Practice**. Tese (Tese de Doutorado em Ciência da Computação) — Université de Liège, Liège, Bélgica, 2014.

MARTÍN-GUTIERREZ, D.; HERNÁNDEZ-PEÑALOZA, G.; BELMONTE-HERNÁNDEZ, A.; ÁLVAREZ, F. A multimodal end-to-end deep learning architecture for music popularity prediction. **IEEE Access**, v. 8, p. 39361–39374, 2020.

PAREEK, P.; SHANKAR, P.; PATHAK, P.; SAKARIYA, N. Predicting music popularity using machine learning algorithm and music metrics available in spotify. **Journal of Development Economics and Management Research Studies**, v. 9, p. 10–19, 2022.

PEDROSO, E.; BORGES, L. T.; OLIVEIRA, P. **Youtube**. Dissertação (Dissertação (Bacharelado em Engenharia)) — Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal, 2016.

RASTELLI, S.; SALLUSTIO, M.; GUARISCO, M. **Spotify and YouTube**. 2022. <https://www.kaggle.com/datasets/salvatorerastelli/spotify-and-youtube>.

TUKEY, J. W. **Exploratory Data Analysis**. Reading, MA, EUA: Addison-Wesley, 1977.

UBC. **Música gravada no Brasil supera, pela primeira vez, os R$ 3 bi em receitas em 2024**. 2025. Accessed: 2025-03-19. Disponível em: <https://tinyurl.com/3fra7yux>.

VASWANI, A.; AL. et. Attention is all you need. In: **Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS)**. Long Beach, CA, EUA: Curran Associates, Inc., 2017. v. 30.

YEE, Y. K.; RAHEEM, M. Predicting music popularity using spotify and youtube features. **Indian Journal of Science and Technology**, v. 15, n. 36, p. 1786–1799, 2022. Disponível em: <https://indjst.org/articles/predicting-music-popularity-using-spotify-and-youtube-features>.

ZERAATGARI, F. Z.; AL. et. Machine learning-based photometric classification of galaxies, quasars, emission-line galaxies and stars. **Monthly Notices of the Royal Astronomical Society**, Oxford University Press, v. 525, n. 1, p. 199–211, 2023.

APÊNDICE A – HIPERPARÂMETROS UTILIZADOS NOS MODELOS

A seguir, são apresentados os hiperparâmetros utilizados para otimização em cada modelo avaliado neste trabalho.

A.1 Random Forest

- **n_estimators**: {300, 400, 500}.
- **max_samples**: {0.5, 0.75}.
- **max_features**: {'sqrt', 'log2'}.
- **max_depth**: {30, 40, 50}.
- **min_samples_split**: {5, 10}.
- **min_samples_leaf**: {2, 4}.

A.2 MLP

- **hidden_layer_sizes**: {(100, 100), (50, 50, 50), (100, 50, 50), (100, 100, 50), (100, 100, 100)}.
- **activation**: {'relu', 'tanh'}.
- **solver**: {'adam', 'sgd'}.
- **learning_rate**: {'constant', 'adaptive'}.
- **alpha**: {0.0001, 0.001}.

A.3 XGBoost

- **n_estimators**: {500, 600, 700, 800, 900, 1000}.
- **learning_rate**: {0.001, 0.01}.
- **gamma**: {0.1, 0.2, 0.3}.
- **min_child_weight**: {1, 2, 3}.
- **subsample**: {0.8, 1.0}.
- **max_depth**: {3, 5, 7, 9}.

A.4 FT-Transformer

- **n_blocks**: {1, . . . , 4}.
- **d_token**: variando de 64 a 512 em passos de 64.
- **attention_n_heads**: {8, 16, 32, 64}.
- **d_ffn_factor**: intervalo contínuo entre 1.0 e 4.0.
- **attention_dropout**: intervalo contínuo entre 0.0 e 0.35, passo 0.05.
- **ffn_dropout**: intervalo contínuo entre 0.0 e 0.5, passo 0.05.
- **residual_dropout**: intervalo contínuo entre 0.0 e 0.2.
- **learning_rate**: intervalo contínuo entre 1e-5 e 1e-3.
- **weight_decay**: intervalo contínuo entre 1e-6 e 1e-3.
- **batch_size**: {128, 256, 512}.
- **n_epochs**: variando de 1000 a 10000 em passos de 200.