Classificação de Distúrbios Vocais Utilizando Aprendizado de Máquina e Análise Espectral

Analaura Rufino de Souza

CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, 2025

Analaura Rufino de Souza

Classificação de Distúrbios Vocais Utilizando Aprendizado de Máquina e Análise Espectral

Monografia apresentada ao curso Ciência da Computação
do Centro de Informática, da Universidade Federal da Paraíba,
como requisito para a obtenção do grau de Bacharel em Ciência da Computação

Orientador: Leonardo Vidal Batista

Maio de 2025

Catalogação na publicação
Seção de Catalogação e Classificação

C614c Souza, Analaura Rufino de.
Classificação de distúrbios vocais utilizando aprendizado de máquina e análise espectral / Analaura Rufino de Souza. - João Pessoa, 2025.
57 f. : il.

Orientação: Leonardo Batista.
TCC (Graduação) - UFPB/Informática.

1. Aprendizado de máquina. 2. Coeficientes Mel. 3. Análise espectral. I. Batista, Leonardo. II. Título.

UFPB/CI CDU 004.8

Elaborado por MICHELLE DE KASSIA FONSECA BARBOSA - CRB-738

CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

Trabalho de Conclusão de Curso de Ciência da Computaçãointitulado ***Classificação de Distúrbios Vocais Utilizando Aprendizado de Máquina e Análise Espectral*** de autoria de Analaura Rufino de Souza, aprovada pela banca examinadora constituída pelos seguintes avaliadores:

Prof. Dr. Leonardo Vidal Batista
Universidade Federal da Paraíba - UFPB

Dr. José Alberto Souza Paulino

Me. Sandoval Verissimo de Sousa Neto

João Pessoa, 9 de maio de 2025

Centro de Informática, Universidade Federal da Paraíba
Rua dos Escoteiros, Mangabeira VII, João Pessoa, Paraíba, Brasil CEP: 58058-600
Fone: +55 (83) 3216 7093 / Fax: +55 (83) 3216 7117

AGRADECIMENTOS

Agradeço, primeiramente, a Deus, por me sustentar em todos os momentos e por me conceder forças, sabedoria e perseverança ao longo dessa jornada.

Aos meus pais, Verinalda e Marcelo, minha base e inspiração, que sempre acreditaram em mim e me apoiaram com amor, paciência e palavras de encorajamento. À minha irmã Luana e ao meu cunhado Junior, que me acolheram durante esse processo de graduação, obrigada pelo incentivo e força, mesmo nos momentos mais difíceis.

Ao professor Leonardo, meu orientador, agradeço sinceramente pela dedicação, orientação e apoio ao longo do desenvolvimento deste trabalho. Sua disponibilidade e conhecimento foram fundamentais para a concretização deste projeto.

Agradeço também ao meu namorado Caio e a todos os meus amigos da União CI, por tornarem esse período mais leve e por compartilharem comigo tantos momentos importantes dessa caminhada.

A todos que, de alguma forma, contribuíram para esta conquista, o meu muito obrigado.

RESUMO

Este trabalho propõe uma abordagem baseada em aprendizado de máquina para a detecção de distúrbios vocais, utilizando características acústicas extraídas dos áudios, como coeficientes Mel-Frequência e parâmetros de perturbação da voz, como *jitter* e *shimmer*. Além disso, foi realizada uma comparação utilizando *data augmentation* para expandir a base de dados e aprimorar a acurácia do modelo. A metodologia adotada divide o sinal em 16 faixas de frequência para a extração das características. Após a extração, cada faixa de frequência é utilizada para treinar um modelo de aprendizado de máquina, totalizando 16 modelos. Os resultados de cada modelo individual são, então, combinados em um meta-modelo, responsável por determinar a classificação final do áudio. No cenário com *data augmentation*, o meta-modelo treinado alcançou uma acurácia de 82,33%, enquanto um dos modelos individuais atingiu 89,16% de acurácia. A abordagem demonstrou ser eficaz, pois, mesmo utilizando menos recursos, superou métodos baseados em redes neurais convolucionais.

Palavras-chave: *aprendizado de máquina*, *coeficientes Mel*, *análise espectral*, *data augmentation*.

ABSTRACT

This work proposes a machine learning-based approach for detecting voice disorders, utilizing acoustic features extracted from audio, such as Mel-Frequency Cepstral Coefficients (MFCCs) and voice perturbation parameters, such as jitter and shimmer. Additionally, a comparison was made using data augmentation to expand the dataset and improve the model's accuracy. The adopted methodology divides the signal into 16 frequency bands for feature extraction. After extraction, each frequency band is used to train a machine learning model, totaling 16 models. The results from each individual model are then combined into a meta-model, responsible for determining the final classification of the audio. In the data augmentation scenario, the trained meta-model achieved an accuracy of 82.33%, while one of the individual models reached an accuracy of 89.16%. The approach proved to be effective, as it surpassed convolutional neural network-based methods, even when using fewer resources.

Keywords: *machine learning*, *Mel coefficients*, *spectral analysis*, *data augmentation.*

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

`ML` – Machine Learning (Aprendizado de Máquina)

`MFCC` – Mel-Frequency Cepstral Coefficients

`CNN` – Convolutional Neural Network (Rede Neural Convolucional)

`ROC` – Receiver Operating Characteristic

`AUC` – Area Under Curve (Área sob a Curva)

`FPR` – False Positive Rate (Taxa de Falsos Positivos)

`TPR` – True Positive Rate (Taxa de Verdadeiros Positivos)

`AUC` – Area Under Curve

Sumário

1 INTRODUÇÃO

1.1 Definição do Problema

A classificação de padrões em sinal de voz é uma área em expansão na computação, com diversas aplicações. Uma de suas aplicações é na área clínica, auxiliando no diagnóstico de doenças. Modelos são treinados para detectar padrões e gerar resultados com base nos parâmetros fornecidos. No entanto, surge a necessidade de desenvolver modelos que apresentem uma alta precisão e, ao mesmo tempo, sejam computacionalmente eficientes.

Uma das técnicas utilizadas para o treinamento é o *machine learning (ML)*. Segundo ORACLE (2025), *machine learning* é um subconjunto da inteligência artificial focado na construção de sistemas que aprendem ou melhoram seu desempenho com base nos dados consumidos. Um dos principais desafios nessa área é definir e extrair as características mais relevantes do sinal de áudio para garantir uma classificação precisa.

1.2 Premissas e Hipóteses

Este trabalho se baseia no estudo feito por (PAULINO et al., 2024), que propôs a divisão de faixas de frequências em espectrogramas, juntamente ao uso de *convolutional neural networks (CNNs)* para a tarefa de classificação de sinais de áudio, alcançando uma acurácia de 82,12%. No entanto, apesar de utilizar a mesma abordagem de divisão de faixas de áudio proposta, o estudo propõe a utilização de características acústicas extraídas diretamente dos sinais de áudio, com foco nos coeficientes de Mel (*Mel-frequency cepstral coefficients* - MFCCs), amplamente utilizados em tarefas de processamento de áudio devido à sua capacidade de representar características perceptivas do som (LOGAN, 2000). Além disso, a aplicação de técnicas de *data augmentation*, como adição de ruído e deslocamento temporal, visa aumentar a robustez do modelo ao lidar com variações nos dados (SHORTEN; KHOSHGOFTAAR, 2019). Para a classificação, modelos de aprendizado de máquina tradicionais, como *Random Forests* ou *Support Vector Machines*, serão explorados, dada sua eficiência em tarefas com características bem definidas (BREIMAN, 2001).

A hipótese defendida é que a combinação da extração de características acústicas, com ênfase nos coeficientes de Mel, a aplicação de técnicas de *data augmentation* e o uso de modelos de aprendizado de máquina tradicionais resultará em um desempenho superior ao das abordagens baseadas em CNNs, em termos de acurácia, eficiência computacional e interpretabilidade do modelo. Essa expectativa é fundamentada na capacidade dos MFCCs de capturar informações acústicas relevantes, na robustez proporcionada pelo

data augmentation e na eficiência de algoritmos de aprendizado de máquina em cenários com características manuais (LOGAN, 2000; SHORTEN; KHOSHGOFTAAR, 2019).

1.3 Objetivo Geral

Desenvolver e avaliar um modelo de aprendizado de máquina para classificação de padrões em sinais de voz, explorando diferentes faixas de frequência e comparando seu desempenho com abordagens baseadas em *CNNs*.

1.4 Objetivos Específicos

1. Extrair características acústicas de sinais de áudio utilizando técnicas de processamento de sinais, incluindo coeficientes Mel-Cepstrais, frequência fundamental (*pitch*), *jitter*, *shimmer* e formantes.

2. Avaliar o impacto das diferentes faixas de frequência na acurácia dos modelos de classificação.

3. Comparar o desempenho de diferentes classificadores de aprendizado de máquina tradicionais e meta-modelagem.

4. Comparar os resultados obtidos com os apresentados no estudo de (PAULINO et al., 2024).

2 CONCEITOS GERAIS

2.1 Amostragem

A amostragem é o processo de conversão de um sinal contínuo em um sinal discreto, capturando valores do sinal em intervalos regulares de tempo. Segundo o Teorema de Nyquist-Shannon, a frequência de amostragem (*sampling rate*) deve ser pelo menos o dobro da maior frequência presente no sinal para evitar o fenômeno de *aliasing*. Em aplicações de áudio, taxas comuns incluem 16 kHz para reconhecimento de fala e 44,1 kHz para aplicações musicais (OPPENHEIM; SCHAFER, 1999).

2.2 Extração de Características Acústicas

A análise de sinais de áudio envolve a extração de diversas características que representam aspectos relevantes da vibração das pregas vocais e da ressonância do trato vocal. O *Praat* permite a extração de vários parâmetros e características do áudio analisado. O presente trabalho concentrou-se na utilização dos seguintes comandos do *Praat* para a extração dessas características:

- *To PointProcess (periodic, cc)*: interpreta um contorno de periodicidade acústica como a frequência de um processo pontual subjacente, como a sequência de fechamentos glotais durante a vibração das pregas vocais (BOERSMA; WEENINK, 2025).
- *Get jitter (local)*: mede a variação relativa ciclo a ciclo do período da onda sonora (BOERSMA; WEENINK, 2025).
- *Get jitter (local, absolute)*: similar ao *local jitter*, mas expressa a variação em valores absolutos (BOERSMA; WEENINK, 2025).
- *Get jitter (rap)*: mede a perturbação relativa média (*Relative Average Perturbation – RAP*), considerando três ciclos consecutivos (BOERSMA; WEENINK, 2025).
- *Get jitter (ppq5)*: mede a perturbação relativa média em cinco ciclos consecutivos, suavizando pequenas variações (BOERSMA; WEENINK, 2025).
- *Get jitter (ddp)*: mede a diferença entre as diferenças consecutivas dos períodos, avaliando variações de curto prazo (BOERSMA; WEENINK, 2025).
- *Get shimmer (local)*: mede a variação na amplitude entre ciclos consecutivos (BOERSMA; WEENINK, 2025).

- *Get shimmer (local dB)*: mede a variação da amplitude em decibéis (BOERSMA; WEENINK, 2025).
- *Get shimmer (apq3)*: mede a perturbação da amplitude considerando três ciclos consecutivos (BOERSMA; WEENINK, 2025).
- *Get shimmer (apq5)*: mede a perturbação da amplitude considerando cinco ciclos consecutivos (BOERSMA; WEENINK, 2025).
- *Get shimmer (apq11)*: mede a perturbação da amplitude considerando onze ciclos consecutivos (BOERSMA; WEENINK, 2025).
- *Get shimmer (dda)*: mede a diferença entre as diferenças consecutivas das amplitudes, avaliando variações de curto prazo (BOERSMA; WEENINK, 2025).

2.3 Coeficientes Cepstrais de Frequência de Mel

Os coeficientes Mel-Cepstrais *(MFCCs)* são características amplamente utilizadas na análise de sinais de áudio, especialmente no reconhecimento de fala e na classificação de padrões acústicos. Esses coeficientes representam a envoltória espectral do sinal de áudio em uma escala perceptiva baseada na forma como o ouvido humano percebe as frequências. A escolha dos MFCCs como características para modelagem acústica deve-se ao fato de que eles capturam informações essenciais da estrutura espectral dos sinais de áudio, sendo robustos para diferentes variações do sinal de fala e ruído.

Matematicamente, os *MFCCs* são obtidos aplicando a Transformada Discreta do Cosseno *(DCT)* sobre os logaritmos das energias filtradas na escala de Mel:

$$c_n = \sum_{m=1}^{M} \log(S_m) \cos\left[n\left(m - \frac{1}{2}\right)\frac{\pi}{M}\right] \tag{1}$$

onde:

- c_n é o n-ésimo coeficiente *MFCC*;
- M é o número total de filtros Mel;
- S_m é a energia do sinal no filtro Mel de índice m.

A Figura 7 ilustra os passos para a extração dos MFCCs, descritos a seguir:

1. **Forma de onda**: O sinal de áudio é representado como uma forma de onda, que descreve a amplitude em função do tempo.

2. **Transformada Discreta de Fourier (*DFT*) e espectro logarítmico**: Aplica-se a Transformada Discreta de Fourier para converter o sinal do domínio do tempo para o domínio da frequência, obtendo o espectro de potência. Em seguida, calcula-se o logaritmo da amplitude espectral, que estabiliza as variações dinâmicas e destaca as características relevantes do sinal (OPPENHEIM; SCHAFER, 1999).

3. **Escala Mel**: O espectro é mapeado para a escala Mel, que reflete a percepção não linear da frequência pelo sistema auditivo humano.

4. **Transformada Discreta de Cosseno (*DCT*)**: A Transformada Discreta de Cosseno é aplicada ao espectro Mel logarítmico, gerando os coeficientes cepstrais (*MFCCs*).

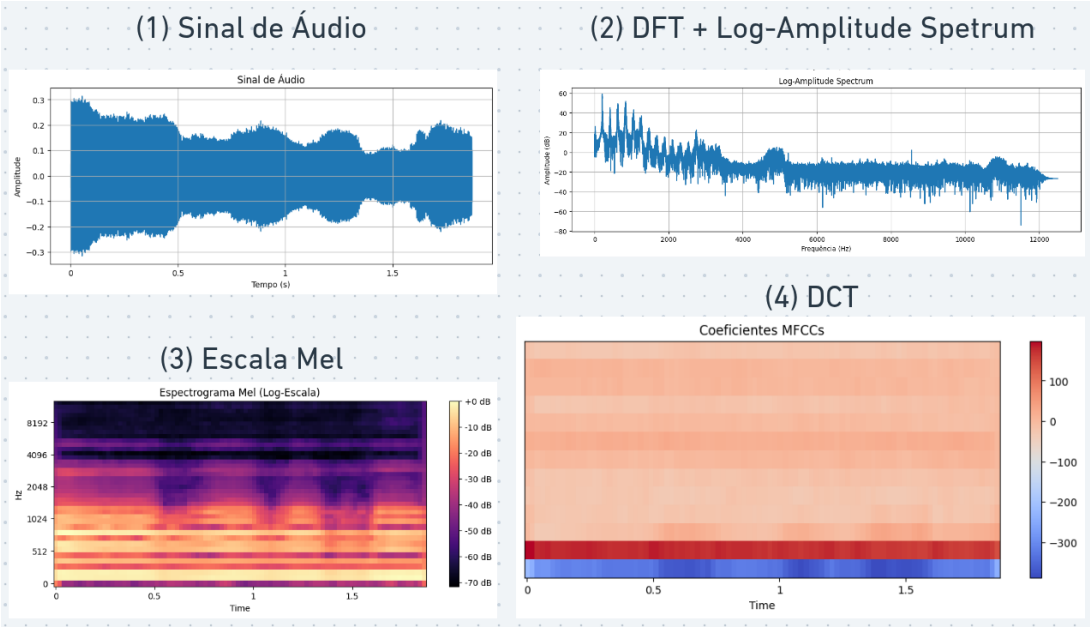

Figura 1: Processo de extração dos *MFCCs* Fonte: Autoria Própria

2.4 Transformada de Fourier e Espectrograma

A *Transformada de Fourier* é uma técnica matemática que permite decompor um sinal no domínio do tempo em suas componentes de frequência. Para sinais discretos, utiliza-se a Transformada Rápida de Fourier (FFT - *Fast Fourier Transform*), que computa a Transformada Discreta de Fourier (DFT - *Discrete Fourier Transform*) de maneira eficiente. A equação da DFT é dada por:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, \quad k = 0, 1, ..., N-1 \tag{2}$$

onde:

- $X[k]$ representa o espectro de frequência do sinal;
- $x[n]$ é o sinal de entrada no tempo discreto;
- N é o número total de amostras;
- j é a unidade imaginária.

Uma limitação da Transformada de Fourier tradicional é que ela não fornece informações sobre como a frequência varia ao longo do tempo. Para contornar essa limitação, utiliza-se a *Transformada de Fourier de Curto Prazo* (STFT - *Short-Time Fourier Transform*), que aplica a FFT em pequenas janelas temporais do sinal:

$$STFT\{x[n]\}(m,k) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j2\pi kn/N} \tag{3}$$

onde $w[n]$ é uma janela deslizante aplicada ao sinal antes da Transformada de Fourier. O resultado da STFT é um espectrograma, uma representação visual da energia do sinal ao longo do tempo e da frequência. Ele é comumente utilizado para análise de padrões acústicos, reconhecimento de fala e detecção de anomalias em sinais de áudio (RABINER; SCHAFER, 2010).

2.5 Sistema Praat

O *Praat* é um software amplamente utilizado para análise, síntese e manipulação de fala e outros sinais acústicos. Desenvolvido por Paul Boersma e David Weenink, do Instituto de Fonética da Universidade de Amsterdã, o *Praat* permite a extração de diversas características acústicas, como frequência fundamental (*pitch*), formantes, *jitter*, *shimmer* e coeficientes Mel-Cepstrais. Essas características são frequentemente empregadas em estudos de fonética, reconhecimento de padrões e análise de distúrbios vocais (BOERSMA; WEENINK, 2025).

2.6 Métricas de Avaliação do Modelo

A avaliação de modelos de aprendizado de máquina é uma parte essencial no desenvolvimento de sistemas de classificação, pois diferentes métricas podem destacar distintos aspectos do desempenho do modelo, especialmente em contextos de classes desbalanceadas ou múltiplos critérios de decisão.

Dentre as principais métricas utilizadas na tarefa de classificação, destacam-se:

1. **Acurácia (*Accuracy*)**: mede a proporção de previsões corretas em relação ao total de amostras. É calculada como:

$$\text{Acurácia} = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{VN} + \text{FP} + \text{FN}} \tag{4}$$

em que VP representa os verdadeiros positivos, VN os verdadeiros negativos, FP os falsos positivos e FN os falsos negativos.

2. **Área sob a Curva ROC (*AUC - ROC*)**: mede a capacidade do modelo em distinguir entre as classes em diferentes limiares de decisão. Quanto maior a AUC, melhor a separação entre as classes.

3. **Revocação (*Recall* ou Sensibilidade)**: avalia a capacidade do modelo em identificar corretamente as amostras positivas, sendo definida como:

$$\text{Revocação} = \frac{\text{VP}}{\text{VP} + \text{FN}} \tag{5}$$

Um alto valor de revocação indica que poucos exemplos positivos foram classificados erroneamente como negativos.

4. **Precisão (*Precision*)**: mede a proporção de previsões positivas que realmente pertencem à classe positiva, sendo calculada por:

$$\text{Precisão} = \frac{\text{VP}}{\text{VP} + \text{FP}} \tag{6}$$

Alta precisão indica que, dentre as instâncias classificadas como positivas, a maioria de fato pertence à classe positiva.

5. ***F1-Score***: combina precisão e revocação em uma única métrica, utilizando a média harmônica entre ambas:

$$F1 = 2 \times \frac{\text{Precisão} \times \text{Revocação}}{\text{Precisão} + \text{Revocação}} \tag{7}$$

Essa métrica é particularmente útil em contextos de desbalanceamento entre classes, pois equilibra os custos dos diferentes tipos de erro.

6. **Kappa de Cohen** (κ): mede a concordância entre as previsões do modelo e os valores reais, ajustando pela concordância esperada ao acaso. É calculado como:

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{8}$$

em que P_o é a concordância observada e P_e é a concordância esperada aleatoriamente.

7. **Coeficiente de Correlação de Matthews *(MCC)***: considera todas as células da matriz de confusão e fornece uma medida mais equilibrada, mesmo em conjuntos de dados desbalanceados. É definido por:

$$MCC = \frac{(\text{VP} \times \text{VN}) - (\text{FP} \times \text{FN})}{\sqrt{(\text{VP} + \text{FP})(\text{VP} + \text{FN})(\text{VN} + \text{FP})(\text{VN} + \text{FN})}} \tag{9}$$

   Ao contrário da acurácia, o *MCC* é mais robusto em cenários com distribuição desigual entre classes.

A escolha apropriada das métricas de avaliação deve considerar não apenas o desempenho global, mas também as particularidades do problema abordado, como o balanceamento entre classes, a natureza dos erros e os objetivos específicos da aplicação. Modelos com bom desempenho em acurácia, por exemplo, podem falhar ao identificar corretamente instâncias raras, o que torna imprescindível a análise de múltiplas métricas para uma avaliação mais robusta (BREIMAN, 2001).

2.7 Meta-modelo

O conceito de meta-modelo refere-se a uma abordagem em que a saída de vários modelos de aprendizado de máquina é combinada para produzir uma decisão final. Esse processo é comumente denominado *ensemble learning* (aprendizado em conjunto) e tem como objetivo melhorar o desempenho geral do sistema, ao reduzir a variabilidade das previsões e aumentar a robustez. A principal vantagem do uso de um meta-modelo é que ele pode combinar as capacidades de diferentes classificadores, integrando suas contribuições para alcançar um desempenho geral mais preciso.

2.8 Data Augmentation

O *data augmentation* (aumento de dados) é uma técnica amplamente utilizada no treinamento de modelos de aprendizado de máquina, especialmente quando há uma quantidade limitada de dados disponível. Essa técnica envolve a aplicação de transformações nos dados originais para gerar novas instâncias, aumentando, assim, a variabilidade do conjunto de treinamento e auxiliando o modelo a generalizar melhor para dados nunca vistos.

Em tarefas de processamento de sinais de voz, o aumento de dados pode ser realizado de diversas maneiras, como modificações na frequência fundamental, distorções temporais (como variações no tempo de reprodução), adição de ruído, translação no tempo e aumento da amplitude. Essas transformações permitem que o modelo aprenda a identificar padrões invariantes, mesmo quando os dados variam em pequenas condições.

2.9 PyCaret

O *PyCaret* é uma biblioteca de aprendizado de máquina em Python que automatiza o processo de criação, ajuste e avaliação de modelos, reduzindo o tempo necessário para experimentação e otimização. Para problemas de classificação, a biblioteca permite avaliar de forma eficiente diversos algoritmos e escolher aquele que apresenta o melhor desempenho com base em métricas como acurácia, F1-score e AUC-ROC (PyCaret, 2025).

Dentre os modelos suportados pelo *PyCaret*, destacam-se os seguintes:

1. *Extra Trees Classifier*: variação do *Random Forest* que melhora a aleatoriedade na escolha dos pontos de divisão, tornando-o menos sensível a ruídos. Possui boa capacidade de generalização, mas pode ser computacionalmente custoso.

2. *Random Forest Classifier*: conjunto de múltiplas árvores de decisão que reduz o risco de *overfitting*. É robusto para dados com muitas variáveis, mas pode ser lento para grandes conjuntos de dados.

3. *Extreme Gradient Boosting (XGBoost)*: um dos algoritmos mais eficientes para classificação, otimizando o aprendizado por meio de *boosting*. É altamente eficaz, mas requer ajuste fino para evitar *overfitting*.

4. *Light Gradient Boosting Machine (LGBM)*: variante do *boosting* que lida bem com grandes volumes de dados e características categóricas. Apresenta alto desempenho, mas pode ser sensível a ruídos.

5. *Gradient Boosting Classifier*: modelo de *boosting* que constrói árvores de forma sequencial para corrigir erros anteriores. Garante alta precisão, porém exige um tempo maior de treinamento.

6. *Ridge Classifier*: classificador linear que utiliza regularização L2 para evitar *overfitting*. Funciona bem em problemas lineares, mas pode ter desempenho inferior em dados não linearmente separáveis.

7. *AdaBoost Classifier*: combinação adaptativa de múltiplos classificadores fracos, geralmente árvores de decisão simples. Resistente a *overfitting*, mas pode ser instável em dados ruidosos.

8. *Linear Discriminant Analysis (LDA)*: modelo estatístico que maximiza a separabilidade entre classes. Funciona bem em dados gaussianos, mas pode ser limitado em dados altamente não lineares.

9. *Decision Tree Classifier*: modelo simples e interpretável, baseado na divisão recursiva dos dados. Fácil de compreender, porém propenso a *overfitting* se não podado adequadamente.

10. *Logistic Regression*: modelo estatístico utilizado para classificação binária. Rápido e eficiente, mas pode não capturar relações complexas entre variáveis.

11. *Quadratic Discriminant Analysis (QDA)*: variante do LDA que permite fronteiras de decisão não lineares. Mais flexível que o LDA, porém pode ser instável em amostras pequenas.

12. *Naive Bayes*: algoritmo probabilístico baseado no Teorema de Bayes. Rápido e eficiente para textos e dados categóricos, mas assume independência entre variáveis, o que pode não refletir a realidade.

13. *K Neighbors Classifier*: classificador baseado na proximidade entre amostras. Simples e eficaz para dados bem distribuídos, mas pode ser lento em grandes volumes de dados.

14. *SVM - Linear Kernel*: classificador baseado na separação por hiperplanos em espaços de alta dimensionalidade. Funciona bem em dados linearmente separáveis, mas pode ser computacionalmente custoso em grandes bases de dados.

15. *Dummy Classifier*: modelo base que realiza previsões aleatórias ou constantes para fins comparativos. Útil como referência, mas sem valor preditivo real.

Dessa forma, o *PyCaret* possibilita a experimentação rápida e estruturada de diferentes abordagens, auxiliando na identificação do modelo mais adequado para um determinado conjunto de dados de classificação.

3 REVISÃO DA LITERATURA

Esta seção apresenta uma revisão de estudos relevantes que exploram diferentes abordagens para análise e classificação de sinais de áudio, destacando suas metodologias, contribuições e limitações.

3.1 Análise de Faixas de Frequência na Detecção de Distúrbios Vocais

A influência das faixas de frequência na classificação de distúrbios vocais foi analisada por (PAULINO et al., 2024), que avaliaram diferentes intervalos espectrais utilizando redes neurais convolucionais (*CNNs*) treinadas com espectrogramas de voz. No estudo, os autores dividiram os sinais de áudio em 16 faixas de frequência e testaram a relevância de cada uma para a classificação de vozes saudáveis e patológicas. Os resultados indicaram que o intervalo entre 1 e 1,462 Hz possui a maior capacidade descritiva em termos de características vocais, sendo significativamente mais eficaz para a análise de patologias vocais. Embora as faixas acima de 5kHz também apresentem relevância para a detecção de padrões patológicos, elas não oferecem a mesma capacidade descritiva em espectrogramas, como observado no estudo.

Além disso, o estudo revelou que, ao combinar regiões de frequências altas com outras faixas, houve uma melhoria no desempenho do modelo de classificação, com a precisão do teste aumentando de 80,53% para 82,10% no dataset SVD e de 78,11% para 82,12% no dataset AVFAD. Esses resultados destacam a importância de uma abordagem cuidadosa na escolha das faixas espectrais e na combinação das mesmas para melhorar a acurácia na classificação de distúrbios vocais.

3.2 Extração de Características Acústicas

A extração de características é um passo crucial na classificação de sinais de áudio. (RAWAT et al., 2023) investigaram a eficácia dos coeficientes Mel-Frequência (*MFCCs*) e da transformada de Fourier de curto prazo (*STFT*) para a tarefa de classificação musical. Eles usaram redes neurais artificiais (*ANNs*) para os *MFCCs* e *CNNs* para espectrogramas, concluindo que a representação visual dos espectrogramas, associada às *CNNs*, foi eficaz na captura de padrões temporais e espectrais, resultando em uma melhor precisão na categorização musical.

Além disso, o estudo de (VIMAL et al., 2021) investigou a classificação de emoções na voz utilizando *MFCCs* como entrada para modelos de aprendizado de máquina. O estudo comparou diferentes algoritmos, incluindo *SVM*, *Random Forest* e árvores de decisão, e encontrou que o *SVM* obteve o melhor desempenho para a tarefa de reconhecimento

emocional. Esse estudo destaca a importância da escolha adequada do modelo de aprendizado de máquina e das características extraídas para a eficácia na classificação de emoções vocais.

3.3 Classificação de Emoções na Voz com o Uso do PRAAT

O software *PRAAT* tem se mostrado uma ferramenta importante para a análise acústica da voz. O estudo de (MAGDIN et al., 2019) utilizou o *PRAAT* para analisar as emoções na voz, extraindo características como formantes, (*pitch*), e parâmetros de perturbação da voz, como *jitter* e *shimmer*. A pesquisa utilizou técnicas estatísticas, como a análise ANOVA, para selecionar as características mais relevantes para a classificação do estado emocional do usuário. A combinação dessas características com modelos de aprendizado de máquina resultou em uma classificação eficaz do estado emocional, demonstrando o potencial do *PRAAT* em aplicações de interação humano-computador.

3.4 Síntese da Revisão

Os estudos revisados destacam a importância da extração de características acústicas na análise e classificação de sinais de áudio. A divisão do sinal em faixas de frequência, conforme demonstrado por (PAULINO et al., 2024), permite uma análise mais detalhada, sendo particularmente útil na detecção de distúrbios vocais. Além disso, as técnicas de extração de características, como os *MFCCs* e o uso do software *PRAAT*, são fundamentais para a precisão dos modelos de classificação, especialmente na análise de emoções na voz.

A combinação de técnicas de aprendizado de máquina, como *SVM* e *Random Forest*, com a extração de características acústicas detalhadas, tem mostrado ser uma abordagem eficaz para a classificação de sinais de áudio, incluindo a detecção de distúrbios vocais e o reconhecimento de emoções. O uso de *PRAAT* permite uma análise detalhada das características vocais e fortalece a robustez dos classificadores. No entanto, desafios como a generalização dos modelos e a influência do ruído ainda precisam ser enfrentados para melhorar a precisão e a robustez dos sistemas de classificação de áudio.

Tabela 1: Resumo de trabalhos relacionados

Autores	Descrição do Estudo	Acurácia
(PAULINO et al., 2024)	Classificação de sinais saudáveis e patológicos por meio da divisão do sinal em 16 faixas de frequência. Treinamento de redes neurais convolucionais (CNNs) com espectrogramas.	82,12%
(RAWAT et al., 2023)	Classificação de gêneros musicais utilizando extração de MFCCs com redes neurais artificiais (ANNs) e espectrogramas com CNNs.	CNN: 97,6% (2 classes), 76,2% (10 classes); ANN: 96,5% (2 classes), 61,4% (10 classes)
(VIMAL et al., 2021)	Classificação de emoções a partir de MFCCs, utilizando algoritmos de aprendizado de máquina.	88,45%
(MAGDIN et al., 2019)	Detecção de emoções com características extraídas via PRAAT. Utilização de ANOVA para seleção de atributos e CNN para classificação.	94%

Fonte: Elaborado pelo autor com base nos estudos citados.
CNN: redes neurais convolucionais; ANN: redes neurais artificiais; ANOVA: Análise de variância

4 METODOLOGIA

A metodologia utilizada neste trabalho foi baseada no artigo de (PAULINO et al., 2024), intitulado *Analysis of Frequency Range Effect on the Detection of Voice Disorder using Convolutional Neural Networks Trained on Spectrogram Images*. Nesse estudo, o autor utiliza imagens de faixas de frequências extraídas de espectrogramas para treinar *CNNs* e classificar áudios em saudáveis e patológicos.

4.1 Base de dados

A base de dados utilizada neste trabalho é a Saarbruecken Voice Database (WOLDERT-JOKISZ, 2007), composta por 766 áudios gravados de indivíduos, sendo metade pertencente à classe saudável e a outra metade à classe patológica. Essa base foi selecionada devido à sua relevância em estudos de classificação de sinais de voz, especialmente para a identificação de condições patológicas (WOLDERT-JOKISZ, 2007). A divisão dos dados para treinamento, validação e teste seguiu a proporção adotada por (PAULINO et al., 2024), com 80% destinados ao treinamento e 20% para teste e validação.

Inicialmente, planejou-se dividir os 20% restantes em partes iguais, com 10% para validação e 10% para teste. Contudo, essa abordagem resultaria em um desequilíbrio entre as classes, com maior quantidade de áudios saudáveis em relação aos patológicos. Para preservar a proporção equilibrada entre as classes, a divisão foi ajustada, resultando em 612 áudios para treinamento (80%), 78 para validação (10,2%) e 76 para teste (9,8%).

Essa reorganização garantiu a representatividade das classes saudável e patológica em todas as etapas do processo.

4.2 Data Augmentation

Buscando aumentar a acurácia do modelo, foi aplicada a técnica de *data augmentation* nos dados separados para treinamento, visando avaliar seu impacto nos resultados. Para cada áudio da base de treinamento, foram geradas duas versões modificadas: uma com aumento de tonalidade (+2 semitons) e outra com adição de ruído gaussiano. Com isso, a base de dados foi ampliada em 1.224 novos áudios (612 com alteração de tonalidade e 612 com ruído gaussiano), totalizando 1.836 áudios. Essa técnica não foi aplicada nos dados de validação e teste, que permaneceram os mesmos, ou seja, 78 áudios para validação e 76 para teste.

4.3 Extração das características

Para que as informações dos áudios fossem utilizadas pelo modelo, foram realizadas duas formas de extração de características: a extração dos Coeficientes Cepstrais de Frequência Mel através do espectrograma e a extração feita diretamente no sinal do áudio (características acústicas).

4.4 Características dos Coeficientes Cepstrais de Frequência Mel

Foi gerado um espectrograma com taxa de amostragem de 25 kHz, dividido em 16 faixas de frequência e com uma normalização para valores entre 0 e 1, conforme proposto por (PAULINO et al., 2024). Desses espectrogramas gerados, foram extraídos 13 *MFCCs* e calculada a média e o desvio padrão desses 13, gerando 26 características utilizadas como entradas para o modelo.

4.5 Características Acústicas

Utilizou-se a biblioteca *Parselmouth* para processar arquivos no formato `.wav`, extraindo os dados necessários. Dentre as informações extraídas, incluiu-se a análise do *pitch* (frequência fundamental da voz), de onde foram obtidas características como a duração do sinal, a média do *pitch* em Hertz e seu desvio padrão. Além disso, foram calculadas diversas medidas de *Jitter* e *Shimmer*, parâmetros comumente utilizados para avaliar a qualidade da voz. Os seguintes comandos foram aplicados para a extração de características acústicas:

1. `To PointProcess (periodic, cc)`

2. `Get jitter (local)`
3. `Get jitter (local, absolute)`
4. `Get jitter (rap)`
5. `Get jitter (ppq5)`
6. `Get jitter (ddp)`
7. `Get shimmer (local)`
8. `Get shimmer (local dB)`
9. `Get shimmer (apq3)`
10. `Get shimmer (apq5)`
11. `Get shimmer (apq11)`
12. `Get shimmer (dda)`

Além das medidas de *jitter* e *shimmer*, foram extraídos os formantes da voz (*F1*, *F2*, *F3* e *F4*), que correspondem às frequências naturais de ressonância do trato vocal. A extração foi realizada a partir dos pulsos glotais detectados no sinal de áudio analisado. O processo foi dividido em quatro etapas: primeiramente, os pontos de pulsação glotal foram identificados utilizando o comando `To PointProcess (periodic, cc)`, em conjunto com `Get number of points`, para determinar o número máximo de pontos; em seguida, os formantes foram estimados por meio do algoritmo de Burg, utilizando o comando `To Formant (burg)`; na terceira etapa, os valores dos formantes foram extraídos especificamente nos instantes correspondentes aos pulsos glotais; por fim, valores inválidos foram removidos, e as médias das frequências de *F1*, *F2*, *F3* e *F4* foram calculadas e utilizadas como características acústicas no modelo.

4.6 Faixas de Frequências

A segmentação das faixas de frequência utilizadas nesta pesquisa foi inspirada na abordagem proposta por (PAULINO et al., 2024), que sugere a seleção de uma região específica do espectrograma com base na relevância das informações presentes nas frequências mais baixas. No estudo, os autores utilizaram apenas as primeiras 60 linhas do espectrograma. Essa escolha resultou em um intervalo de 1 Hz a, aproximadamente, 1,462 Hz, determinado a partir do teorema de Nyquist-Shannon, que estabelece que a frequência máxima representável em um sinal digital é metade da taxa de amostragem.

Seguindo essa lógica, a segmentação das frequências foi realizada em intervalos fixos de 1,462 Hz, com uma sobreposição de 50%, correspondente ao primeiro intervalo de frequência (1,462 Hz), e a diferença entre os termos consecutivos é 731 Hz. Dessa forma, foram geradas 16 faixas de frequência, cobrindo toda a faixa espectral de interesse até 12,5 kHz. Contudo, devido a restrições impostas pelo algoritmo de análise de *pitch* do *Praat*, na etapa de processamento direto do áudio, a primeira faixa de frequência foi ajustada de 1 Hz para 8 Hz. O software determina a janela de análise com base na menor frequência permitida *(Pitch Floor)* e na taxa de amostragem do áudio, para que a janela contenha pelo menos três períodos completos da frequência mínima. Quando esse valor é muito baixo, como 1 Hz, a janela de análise se torna longa, gerando erros na detecção do *pitch*. Nos áudios da base de dados utilizada, valores inferiores a 8 Hz resultavam em falhas na extração do *pitch*, tornando necessária essa adaptação para extrair os dados da primeira faixa. Dessa forma, as faixas de frequência para essa etapa ajustada são:

Tabela 2: Intervalos de frequências

Faixa	Frequência Inicial	Frequência Final
1	8	1,462
2	731	2,193
3	1,463	2,924
4	2,193	3,655
5	2,924	4,386
6	3,655	5,117
7	4,386	5,848
8	5,117	6,579
9	5,848	7,310
10	6579	8041
11	7310	8772
12	8041	9503
13	8772	10234
14	9503	10965
15	10234	11696
16	10965	12427

4.7 Tratamento dos dados

As características acústicas foram extraídas apenas na primeira faixa de áudio (8 Hz a 1,462 Hz). Como a voz humana pode variar entre, aproximadamente, 80 Hz e 450 Hz (TITZE, 1994), as demais faixas de áudio não abrangeriam essas frequências, e não seria possível calcular o *pitch* e as características acústicas. Portanto, a primeira faixa foi utilizada tanto para a extração das características acústicas quanto dos coeficientes de Mel, enquanto as outras 15 faixas seguintes foram utilizadas apenas para a extração dos coeficientes de Mel.

Com isso, foi realizada a extração dos dados de todos os áudios da base de dados. Para cada faixa, os valores extraídos foram organizados em uma estrutura tabular, em que cada linha representa um áudio, e cada coluna corresponde a uma característica extraída. Além disso, foi adicionada uma coluna indicando a classe do áudio (saudável ou patológico), a qual foi utilizada posteriormente no treinamento do modelo.

4.8 Treinamento

Para cada conjunto de dados correspondente a uma faixa de frequência, foi criada uma instância na biblioteca *PyCaret* passando como parâmetros os dados de treinamento no campo `data`, os dados de teste no campo `test_data`, a coluna `class` no campo `target` e a coluna `file_path` como parâmetro de exclusão (`ignore_features`), já que

representava apenas o caminho do arquivo. O restante dos parâmetros seguiu o padrão da biblioteca. Ao iniciar, a biblioteca realiza uma comparação entre os seguintes modelos de classificação: *Extra Trees Classifier, Random Forest Classifier, Extreme Gradient Boosting, Light Gradient Boosting Machine, Gradient Boosting Classifier, Ridge Classifier, Ada Boost Classifier, Linear Discriminant Analysis, Decision Tree Classifier, Logistic Regression, Quadratic Discriminant Analysis, Naive Bayes, K Neighbors Classifier, SVM - Linear Kernel* e *Dummy Classifier*. Para selecionar o modelo final, a biblioteca utiliza a métrica acurácia; esse modelo escolhido será utilizado na próxima etapa de classificação final (PYCARET, 2025).

4.9 Classificação Final

Como foi realizada uma classificação para cada faixa de frequência, tornou-se necessário consolidar essas predições em um único resultado final para cada áudio. Para isso, duas abordagens foram aplicadas. A primeira consistiu em uma votação majoritária, na qual a classe mais frequentemente predita entre as faixas era escolhida como resultado final. A segunda abordagem envolveu o treinamento de um meta-modelo, com o objetivo de aprender quais faixas de frequência possuem maior influência na decisão final. Para isso, foram utilizadas as predições geradas pelos modelos individuais selecionados na etapa anterior, aplicadas sobre a base de validação. Cada predição foi registrada como uma coluna em uma nova tabela, onde cada linha correspondia a um áudio. Adicionalmente, foi inserida uma coluna-alvo com a classe real de cada áudio. Essa nova base de dados, composta pelas predições dos modelos individuais (como *features*) e pelos respectivos rótulos reais, foi então utilizada para o treinamento do meta-modelo. O *PyCaret* foi utilizado para esse processo, dividindo automaticamente os dados da base de validação em 70% para treinamento e 30% para validação, conforme sua configuração padrão (PYCARET, 2025).

5 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Este capítulo apresenta os resultados obtidos a partir da aplicação da metodologia descrita anteriormente. Para avaliar o desempenho de cada modelo, foram utilizadas as métricas fornecidas pela biblioteca *PyCaret* (PyCaret, 2025), apresentadas em forma de tabela, a qual exibe os modelos testados juntamente com seus respectivos resultados. As tabelas estão ordenadas com base na acurácia dos modelos.

Além disso, foram construídos dois gráficos para o modelo selecionado: a curva ROC e a matriz de confusão, ambos utilizando os dados de teste. Por uma questão de organização, esses gráficos foram apresentados apenas para o melhor modelo individual e para o meta-modelo. O modelo escolhido para a segunda etapa do treinamento é sempre aquele que apresenta a maior acurácia entre os avaliados.

Para melhor visualização, os resultados foram divididos em dois cenários distintos: sem aplicação de *data augmentation* e com aplicação de *data augmentation*.

5.1 Resultados sem Data Augmentation

A avaliação inicial dos modelos foi realizada sem a aplicação de técnicas de *data augmentation*. Os resultados obtidos para cada modelo são apresentados nas tabelas e figuras a seguir.

Tabela 3: Resultados do modelo – Faixa 1

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Extreme Gradient Boosting	0.7419	0.8252	0.7419	0.7467	0.7408	0.4837	0.4884	0.3470
Random Forest Classifier	0.7370	0.8255	0.7370	0.7408	0.7358	0.4739	0.4776	0.3890
Gradient Boosting Classifier	0.7370	0.8213	0.7370	0.7425	0.7354	0.4738	0.4793	0.7990
Light Gradient Boosting Machine	0.7354	0.8224	0.7354	0.7399	0.7343	0.4708	0.4752	0.5450
Ridge Classifier	0.7289	0.8094	0.7289	0.7355	0.7268	0.4579	0.4643	0.0360
Extra Trees Classifier	0.7288	0.8250	0.7288	0.7341	0.7271	0.4576	0.4627	0.4010
Logistic Regression	0.7223	0.7831	0.7223	0.7258	0.7212	0.4446	0.4480	0.2310
Linear Discriminant Analysis	0.7125	0.8083	0.7125	0.7197	0.7103	0.4252	0.4321	0.0490
Ada Boost Classifier	0.7124	0.7824	0.7124	0.7182	0.7100	0.4253	0.4308	0.3690
Quadratic Discriminant Analysis	0.7027	0.7933	0.7027	0.7490	0.6892	0.4053	0.4488	0.0600
Naive Bayes	0.6848	0.8047	0.6848	0.7622	0.6589	0.3692	0.4388	0.0330
Decision Tree Classifier	0.6846	0.6846	0.6846	0.6862	0.6839	0.3691	0.3707	0.0490
K Neighbors Classifier	0.5934	0.6256	0.5934	0.5958	0.5912	0.1873	0.1893	0.0380
SVM - Linear Kernel	0.5407	0.6328	0.5407	0.5828	0.4209	0.0756	0.1424	0.0400
Dummy Classifier	0.4934	0.5000	0.4934	0.2435	0.3261	0.0000	0.0000	0.0570

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 4: Resultados do modelo – Faixa 2

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Extra Trees Classifier	0.7482	0.8260	0.7482	0.7506	0.7476	0.4965	0.4988	0.3090
Random Forest Classifier	0.7467	0.8313	0.7467	0.7497	0.7459	0.4933	0.4963	0.6600
Gradient Boosting Classifier	0.7451	0.8153	0.7451	0.7473	0.7444	0.4899	0.4922	0.5900
Naive Bayes	0.7401	0.8048	0.7401	0.7420	0.7394	0.4799	0.4819	0.0350
Light Gradient Boosting Machine	0.7386	0.8153	0.7386	0.7400	0.7383	0.4772	0.4785	0.3130
Ridge Classifier	0.7352	0.8082	0.7352	0.7397	0.7337	0.4704	0.4748	0.0320
Linear Discriminant Analysis	0.7352	0.8084	0.7352	0.7397	0.7337	0.4704	0.4748	0.0570
Logistic Regression	0.7238	0.8070	0.7238	0.7280	0.7224	0.4475	0.4516	0.1480
Extreme Gradient Boosting	0.7141	0.8105	0.7141	0.7158	0.7136	0.4282	0.4298	0.2670
Ada Boost Classifier	0.7140	0.7971	0.7140	0.7160	0.7134	0.4277	0.4297	0.2090
Quadratic Discriminant Analysis	0.7125	0.8007	0.7125	0.7197	0.7101	0.4246	0.4318	0.0360
Decision Tree Classifier	0.6585	0.6587	0.6585	0.6608	0.6577	0.3171	0.3192	0.0420
SVM - Linear Kernel	0.6535	0.7258	0.6535	0.6774	0.6335	0.3060	0.3269	0.0370
K Neighbors Classifier	0.6292	0.6783	0.6292	0.6346	0.6256	0.2587	0.2638	0.0350
Dummy Classifier	0.4934	0.5000	0.4934	0.2435	0.3261	0.0000	0.0000	0.0290

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews; F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 5: Resultados do modelo – Faixa 3

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Extra Trees Classifier	0.7353	0.7965	0.7353	0.7392	0.7344	0.4701	0.4741	0.2090
Random Forest Classifier	0.7238	0.7907	0.7238	0.7283	0.7226	0.4473	0.4519	0.3600
Gradient Boosting Classifier	0.7108	0.7724	0.7108	0.7174	0.7085	0.4210	0.4277	0.5550
Logistic Regression	0.7011	0.7579	0.7011	0.7059	0.6993	0.4015	0.4065	0.1660
Ridge Classifier	0.6994	0.7597	0.6994	0.7036	0.6978	0.3982	0.4026	0.0330
Linear Discriminant Analysis	0.6994	0.7599	0.6994	0.7036	0.6978	0.3982	0.4026	0.0340
Light Gradient Boosting Machine	0.6978	0.7641	0.6978	0.7021	0.6962	0.3953	0.3996	0.4770
Quadratic Discriminant Analysis	0.6976	0.7518	0.6976	0.7112	0.6930	0.3948	0.4078	0.0650
Naive Bayes	0.6960	0.7868	0.6960	0.7035	0.6934	0.3914	0.3989	0.0360
Extreme Gradient Boosting	0.6928	0.7535	0.6928	0.6961	0.6914	0.3853	0.3886	0.5960
Ada Boost Classifier	0.6750	0.7186	0.6750	0.6777	0.6738	0.3499	0.3526	0.3170
K Neighbors Classifier	0.6293	0.6689	0.6293	0.6348	0.6266	0.2591	0.2642	0.0370
Decision Tree Classifier	0.6045	0.6042	0.6045	0.6057	0.6022	0.2084	0.2098	0.0420
SVM - Linear Kernel	0.5539	0.7161	0.5539	0.5873	0.4578	0.1031	0.1587	0.0350
Dummy Classifier	0.4934	0.5000	0.4934	0.2435	0.3261	0.0000	0.0000	0.0300

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews; F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 6: Resultados do modelo – Faixa 4

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Random Forest Classifier	0.7319	0.7993	0.7319	0.7376	0.7305	0.4637	0.4693	0.3350
Gradient Boosting Classifier	0.7221	0.7751	0.7221	0.7274	0.7208	0.4440	0.4493	0.7040
Light Gradient Boosting Machine	0.7140	0.7786	0.7140	0.7234	0.7118	0.4280	0.4370	0.6660
Naive Bayes	0.7058	0.7748	0.7058	0.7170	0.7020	0.4113	0.4223	0.0310
Ridge Classifier	0.7026	0.7572	0.7026	0.7067	0.7016	0.4053	0.4092	0.0330
Linear Discriminant Analysis	0.7026	0.7572	0.7026	0.7067	0.7016	0.4053	0.4092	0.0320
Logistic Regression	0.6994	0.7555	0.6994	0.7021	0.6986	0.3988	0.4014	0.2470
Extra Trees Classifier	0.6993	0.7894	0.6993	0.7031	0.6979	0.3983	0.4021	0.2160
Extreme Gradient Boosting	0.6944	0.7707	0.6944	0.6990	0.6927	0.3885	0.3932	0.2550
Ada Boost Classifier	0.6845	0.7470	0.6845	0.6873	0.6833	0.3691	0.3717	0.2280
Decision Tree Classifier	0.6421	0.6420	0.6421	0.6446	0.6409	0.2842	0.2866	0.0410
Quadratic Discriminant Analysis	0.6389	0.7068	0.6389	0.6599	0.6284	0.2775	0.2975	0.0310
SVM - Linear Kernel	0.6195	0.7025	0.6195	0.6715	0.5741	0.2366	0.2794	0.0410
K Neighbors Classifier	0.5897	0.6224	0.5897	0.5908	0.5883	0.1789	0.1802	0.0500
Dummy Classifier	0.4934	0.5000	0.4934	0.2435	0.3261	0.0000	0.0000	0.0280

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 7: Resultados do modelo – Faixa 5

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Random Forest Classifier	0.7321	0.7878	0.7321	0.7348	0.7313	0.4639	0.4667	0.3460
Light Gradient Boosting Machine	0.7321	0.7758	0.7321	0.7348	0.7313	0.4642	0.4668	0.6770
Gradient Boosting Classifier	0.7191	0.7893	0.7191	0.7215	0.7184	0.4382	0.4406	0.7280
Extra Trees Classifier	0.7191	0.7935	0.7191	0.7220	0.7184	0.4381	0.4410	0.2080
Naive Bayes	0.7139	0.7812	0.7139	0.7285	0.7093	0.4277	0.4419	0.0590
Extreme Gradient Boosting	0.7060	0.7738	0.7060	0.7077	0.7056	0.4120	0.4137	0.2460
Ridge Classifier	0.6995	0.7671	0.6995	0.7010	0.6991	0.3993	0.4006	0.0320
Linear Discriminant Analysis	0.6995	0.7670	0.6995	0.7010	0.6991	0.3993	0.4006	0.0340
Logistic Regression	0.6994	0.7670	0.6994	0.7009	0.6991	0.3990	0.4003	0.2080
Ada Boost Classifier	0.6863	0.7477	0.6863	0.6882	0.6850	0.3720	0.3741	0.2010
Decision Tree Classifier	0.6422	0.6422	0.6422	0.6444	0.6411	0.2844	0.2865	0.0720
Quadratic Discriminant Analysis	0.6404	0.7335	0.6404	0.6680	0.6207	0.2807	0.3064	0.0330
K Neighbors Classifier	0.6244	0.6534	0.6244	0.6256	0.6232	0.2487	0.2498	0.0640
SVM - Linear Kernel	0.5475	0.7078	0.5475	0.6223	0.4719	0.0967	0.1418	0.0370
Dummy Classifier	0.4934	0.5000	0.4934	0.2435	0.3261	0.0000	0.0000	0.0260

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 8: Resultados do modelo – Faixa 6

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Random Forest Classifier	0.7026	0.7494	0.7026	0.7050	0.7015	0.4052	0.4075	0.3560
Extra Trees Classifier	0.6765	0.7383	0.6765	0.6777	0.6760	0.3530	0.3542	0.2070
Gradient Boosting Classifier	0.6519	0.7173	0.6519	0.6535	0.6509	0.3037	0.3053	0.6950
Light Gradient Boosting Machine	0.6503	0.7190	0.6503	0.6527	0.6487	0.3007	0.3030	0.7600
Logistic Regression	0.6487	0.7047	0.6487	0.6495	0.6481	0.2972	0.2980	0.1580
Ridge Classifier	0.6487	0.7044	0.6487	0.6493	0.6481	0.2971	0.2978	0.0560
Linear Discriminant Analysis	0.6487	0.7042	0.6487	0.6493	0.6481	0.2971	0.2978	0.0430
Extreme Gradient Boosting	0.6421	0.7081	0.6421	0.6458	0.6393	0.2843	0.2878	0.2650
Naive Bayes	0.6405	0.7156	0.6405	0.6601	0.6295	0.2806	0.2991	0.0340
Ada Boost Classifier	0.6325	0.6727	0.6325	0.6347	0.6306	0.2647	0.2670	0.2050
Decision Tree Classifier	0.6261	0.6261	0.6261	0.6282	0.6247	0.2522	0.2542	0.0800
Quadratic Discriminant Analysis	0.6046	0.6756	0.6046	0.6272	0.5877	0.2089	0.2301	0.0350
K Neighbors Classifier	0.5721	0.5892	0.5721	0.5756	0.5693	0.1446	0.1477	0.0360
SVM - Linear Kernel	0.5654	0.6643	0.5654	0.6069	0.4914	0.1289	0.1703	0.0630
Dummy Classifier	0.4934	0.5000	0.4934	0.2435	0.3261	0.0000	0.0000	0.0310

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 9: Resultados do modelo – Faixa 7

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Extra Trees Classifier	0.6666	0.7326	0.6666	0.6681	0.6658	0.3328	0.3344	0.2220
Random Forest Classifier	0.6616	0.7233	0.6616	0.6649	0.6601	0.3235	0.3265	0.3970
Light Gradient Boosting Machine	0.6456	0.7044	0.6456	0.6468	0.6451	0.2913	0.2924	0.7400
Extreme Gradient Boosting	0.6455	0.6889	0.6455	0.6485	0.6435	0.2909	0.2938	0.2700
Gradient Boosting Classifier	0.6371	0.6940	0.6371	0.6394	0.6361	0.2743	0.2764	0.7470
Naive Bayes	0.6323	0.6989	0.6323	0.6430	0.6237	0.2646	0.2746	0.0320
Logistic Regression	0.6291	0.6898	0.6291	0.6324	0.6271	0.2590	0.2618	0.1870
Ridge Classifier	0.6291	0.6878	0.6291	0.6325	0.6271	0.2593	0.2620	0.0610
Linear Discriminant Analysis	0.6291	0.6879	0.6291	0.6325	0.6271	0.2593	0.2620	0.0370
Ada Boost Classifier	0.6210	0.6532	0.6210	0.6226	0.6198	0.2420	0.2435	0.2080
Quadratic Discriminant Analysis	0.5979	0.6681	0.5979	0.6117	0.5845	0.1959	0.2090	0.0360
K Neighbors Classifier	0.5587	0.5622	0.5587	0.5595	0.5563	0.1163	0.1175	0.0370
SVM - Linear Kernel	0.5524	0.6366	0.5524	0.5635	0.4703	0.1064	0.1403	0.0690
Decision Tree Classifier	0.5473	0.5477	0.5473	0.5490	0.5455	0.0953	0.0966	0.0690
Dummy Classifier	0.4934	0.5000	0.4934	0.2435	0.3261	0.0000	0.0000	0.0290

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 10: Resultados do modelo – Faixa 8

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Random Forest Classifier	0.6601	0.7245	0.6601	0.6626	0.6588	0.3203	0.3226	0.3910
Logistic Regression	0.6585	0.7005	0.6585	0.6615	0.6572	0.3171	0.3200	0.1910
Extra Trees Classifier	0.6552	0.7036	0.6552	0.6587	0.6530	0.3102	0.3136	0.2200
Extreme Gradient Boosting	0.6536	0.6963	0.6536	0.6564	0.6524	0.3079	0.3102	0.2660
Ridge Classifier	0.6504	0.6998	0.6504	0.6529	0.6493	0.3009	0.3032	0.0610
Linear Discriminant Analysis	0.6504	0.6996	0.6504	0.6529	0.6493	0.3009	0.3032	0.0530
Light Gradient Boosting Machine	0.6406	0.7003	0.6406	0.6420	0.6396	0.2810	0.2824	0.7730
Gradient Boosting Classifier	0.6356	0.6881	0.6356	0.6370	0.6348	0.2713	0.2725	0.6940
Naive Bayes	0.6292	0.6962	0.6292	0.6325	0.6269	0.2587	0.2617	0.0310
Quadratic Discriminant Analysis	0.6160	0.6537	0.6160	0.6223	0.6111	0.2323	0.2382	0.0330
Decision Tree Classifier	0.6078	0.6082	0.6078	0.6102	0.6066	0.2162	0.2182	0.0620
Ada Boost Classifier	0.5851	0.6260	0.5851	0.5861	0.5844	0.1706	0.1713	0.2020
SVM - Linear Kernel	0.5360	0.6805	0.5360	0.6401	0.4263	0.0663	0.1322	0.0660
Dummy Classifier	0.4934	0.5000	0.4934	0.2435	0.3261	0.0000	0.0000	0.0300
K Neighbors Classifier	0.4916	0.5016	0.4916	0.4922	0.4895	-0.0167	-0.0163	0.0380

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 11: Resultados do modelo – Faixa 9

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Random Forest Classifier	0.7010	0.7212	0.7010	0.7057	0.6992	0.4015	0.4063	0.3530
Extra Trees Classifier	0.6765	0.7322	0.6765	0.6789	0.6751	0.3526	0.3552	0.2230
Naive Bayes	0.6585	0.7213	0.6585	0.6606	0.6576	0.3168	0.3189	0.0340
Ridge Classifier	0.6553	0.7210	0.6553	0.6589	0.6533	0.3107	0.3141	0.0560
Linear Discriminant Analysis	0.6553	0.7209	0.6553	0.6589	0.6533	0.3107	0.3141	0.0530
Logistic Regression	0.6536	0.7207	0.6536	0.6570	0.6516	0.3072	0.3105	0.1840
Gradient Boosting Classifier	0.6536	0.7071	0.6536	0.6565	0.6513	0.3065	0.3096	0.6960
Light Gradient Boosting Machine	0.6471	0.6984	0.6471	0.6487	0.6458	0.2940	0.2956	0.5530
Extreme Gradient Boosting	0.6405	0.6864	0.6405	0.6437	0.6380	0.2809	0.2841	0.2640
Ada Boost Classifier	0.6306	0.6620	0.6306	0.6318	0.6295	0.2612	0.2623	0.2010
Quadratic Discriminant Analysis	0.6209	0.6613	0.6209	0.6260	0.6166	0.2422	0.2469	0.0330
SVM - Linear Kernel	0.5882	0.7087	0.5882	0.6874	0.5211	0.1766	0.2401	0.0670
Decision Tree Classifier	0.5326	0.5325	0.5326	0.5326	0.5303	0.0650	0.0650	0.0760
Dummy Classifier	0.4934	0.5000	0.4934	0.2435	0.3261	0.0000	0.0000	0.0510
K Neighbors Classifier	0.4790	0.4881	0.4790	0.4788	0.4767	-0.0426	-0.0426	0.0370

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 12: Resultados do modelo – Faixa 10

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Naive Bayes	0.6748	0.7193	0.6748	0.6768	0.6737	0.3491	0.3513	0.0310
Extra Trees Classifier	0.6700	0.7296	0.6700	0.6727	0.6686	0.3393	0.3423	0.2880
Quadratic Discriminant Analysis	0.6636	0.7099	0.6636	0.6682	0.6600	0.3264	0.3313	0.0370
Random Forest Classifier	0.6585	0.7206	0.6585	0.6640	0.6569	0.3166	0.3222	0.5070
Logistic Regression	0.6503	0.7010	0.6503	0.6532	0.6490	0.3002	0.3032	0.1990
Ridge Classifier	0.6470	0.6976	0.6470	0.6502	0.6453	0.2936	0.2969	0.0560
Linear Discriminant Analysis	0.6470	0.6976	0.6470	0.6502	0.6453	0.2936	0.2969	0.0620
Extreme Gradient Boosting	0.6438	0.6951	0.6438	0.6460	0.6428	0.2878	0.2898	0.2700
Gradient Boosting Classifier	0.6342	0.6913	0.6342	0.6380	0.6324	0.2684	0.2721	0.5980
Light Gradient Boosting Machine	0.6226	0.6733	0.6226	0.6246	0.6210	0.2449	0.2469	0.4000
Ada Boost Classifier	0.5866	0.6164	0.5866	0.5889	0.5839	0.1734	0.1755	0.2150
Decision Tree Classifier	0.5638	0.5630	0.5638	0.5645	0.5608	0.1261	0.1275	0.0420
SVM - Linear Kernel	0.5409	0.6787	0.5409	0.6136	0.4339	0.0780	0.1540	0.0370
K Neighbors Classifier	0.5181	0.5089	0.5181	0.5188	0.5143	0.0357	0.0365	0.0360
Dummy Classifier	0.4934	0.5000	0.4934	0.2435	0.3261	0.0000	0.0000	0.0300

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 13: Resultados do modelo – Faixa 11

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Naive Bayes	0.6536	0.7038	0.6536	0.6572	0.6514	0.3064	0.3103	0.0330
Ridge Classifier	0.6356	0.6917	0.6356	0.6384	0.6336	0.2708	0.2737	0.0320
Linear Discriminant Analysis	0.6356	0.6917	0.6356	0.6384	0.6336	0.2708	0.2737	0.0320
Logistic Regression	0.6340	0.6939	0.6340	0.6370	0.6315	0.2673	0.2705	0.1970
Extra Trees Classifier	0.6290	0.6897	0.6290	0.6310	0.6269	0.2569	0.2594	0.2490
Random Forest Classifier	0.6276	0.6934	0.6276	0.6301	0.6255	0.2543	0.2571	0.4090
Extreme Gradient Boosting	0.6226	0.6566	0.6226	0.6244	0.6213	0.2446	0.2466	0.4200
Light Gradient Boosting Machine	0.6210	0.6680	0.6210	0.6235	0.6188	0.2414	0.2441	0.3090
Gradient Boosting Classifier	0.6111	0.6651	0.6111	0.6139	0.6092	0.2216	0.2246	0.5400
Ada Boost Classifier	0.6094	0.6676	0.6094	0.6107	0.6074	0.2191	0.2201	0.2570
Quadratic Discriminant Analysis	0.6013	0.6651	0.6013	0.6055	0.5979	0.2024	0.2065	0.0620
SVM - Linear Kernel	0.5344	0.6579	0.5344	0.5977	0.4208	0.0650	0.1244	0.0370
Decision Tree Classifier	0.5312	0.5312	0.5312	0.5327	0.5288	0.0625	0.0638	0.0430
K Neighbors Classifier	0.5214	0.5398	0.5214	0.5220	0.5196	0.0432	0.0435	0.0360
Dummy Classifier	0.4934	0.5000	0.4934	0.2435	0.3261	0.0000	0.0000	0.0330

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 14: Resultados do modelo – Faixa 12

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Random Forest Classifier	0.6553	0.6828	0.6553	0.6586	0.6528	0.3099	0.3134	0.3450
Gradient Boosting Classifier	0.6338	0.6728	0.6338	0.6369	0.6312	0.2669	0.2702	0.7320
Naive Bayes	0.6290	0.6915	0.6290	0.6313	0.6271	0.2571	0.2597	0.0570
Extra Trees Classifier	0.6193	0.6712	0.6193	0.6216	0.6179	0.2379	0.2404	0.2340
Light Gradient Boosting Machine	0.6144	0.6670	0.6144	0.6171	0.6118	0.2278	0.2308	0.6240
Ridge Classifier	0.6078	0.6587	0.6078	0.6096	0.6063	0.2154	0.2172	0.0370
Linear Discriminant Analysis	0.6078	0.6588	0.6078	0.6096	0.6063	0.2154	0.2172	0.0350
Logistic Regression	0.6062	0.6595	0.6062	0.6085	0.6044	0.2123	0.2146	0.2320
Extreme Gradient Boosting	0.6013	0.6507	0.6013	0.6035	0.5989	0.2022	0.2045	0.2730
Ada Boost Classifier	0.5816	0.6136	0.5816	0.5826	0.5803	0.1628	0.1638	0.2060
Quadratic Discriminant Analysis	0.5654	0.6054	0.5654	0.5696	0.5613	0.1306	0.1348	0.0340
Decision Tree Classifier	0.5523	0.5524	0.5523	0.5537	0.5510	0.1047	0.1060	0.0610
SVM - Linear Kernel	0.5359	0.6398	0.5359	0.6168	0.4591	0.0721	0.1185	0.0390
K Neighbors Classifier	0.5131	0.4995	0.5131	0.5140	0.5081	0.0247	0.0262	0.0660
Dummy Classifier	0.4934	0.5000	0.4934	0.2435	0.3261	0.0000	0.0000	0.0270

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 15: Resultados do modelo – Faixa 13

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Random Forest Classifier	0.6634	0.7156	0.6634	0.6680	0.6602	0.3261	0.3309	0.3380
Extra Trees Classifier	0.6585	0.7206	0.6585	0.6622	0.6559	0.3166	0.3203	0.2150
Gradient Boosting Classifier	0.6453	0.6845	0.6453	0.6481	0.6439	0.2905	0.2933	0.7410
Naive Bayes	0.6404	0.6787	0.6404	0.6467	0.6366	0.2808	0.2869	0.0530
Light Gradient Boosting Machine	0.6357	0.6885	0.6357	0.6370	0.6345	0.2713	0.2726	0.6680
Ada Boost Classifier	0.6355	0.6851	0.6355	0.6367	0.6341	0.2705	0.2719	0.2080
Extreme Gradient Boosting	0.6275	0.6790	0.6275	0.6298	0.6256	0.2548	0.2571	0.2660
Logistic Regression	0.6224	0.6709	0.6224	0.6274	0.6195	0.2446	0.2495	0.1910
Ridge Classifier	0.6208	0.6670	0.6208	0.6247	0.6179	0.2411	0.2451	0.0320
Linear Discriminant Analysis	0.6208	0.6668	0.6208	0.6247	0.6179	0.2411	0.2451	0.0350
Quadratic Discriminant Analysis	0.6029	0.6384	0.6029	0.6058	0.6004	0.2066	0.2090	0.0340
Decision Tree Classifier	0.5963	0.5967	0.5963	0.5992	0.5939	0.1931	0.1956	0.0770
K Neighbors Classifier	0.5668	0.5723	0.5668	0.5673	0.5659	0.1332	0.1338	0.0540
SVM - Linear Kernel	0.5098	0.6497	0.5098	0.3882	0.3602	0.0252	0.0515	0.0640
Dummy Classifier	0.4934	0.5000	0.4934	0.2435	0.3261	0.0000	0.0000	0.0350

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 16: Resultados do modelo – Faixa 14

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Random Forest Classifier	0.6537	0.6761	0.6537	0.6565	0.6519	0.3075	0.3102	0.3500
Naive Bayes	0.6488	0.6835	0.6488	0.6531	0.6454	0.2976	0.3018	0.0590
Extra Trees Classifier	0.6438	0.6882	0.6438	0.6459	0.6426	0.2874	0.2895	0.2280
Gradient Boosting Classifier	0.6388	0.6607	0.6388	0.6416	0.6373	0.2779	0.2805	0.7370
Light Gradient Boosting Machine	0.6307	0.6612	0.6307	0.6331	0.6287	0.2613	0.2636	0.6660
Extreme Gradient Boosting	0.6290	0.6737	0.6290	0.6318	0.6269	0.2582	0.2608	0.2760
Ridge Classifier	0.6225	0.6677	0.6225	0.6253	0.6209	0.2455	0.2479	0.0320
Linear Discriminant Analysis	0.6225	0.6679	0.6225	0.6253	0.6209	0.2455	0.2479	0.0360
Logistic Regression	0.6209	0.6690	0.6209	0.6238	0.6193	0.2422	0.2448	0.2020
Ada Boost Classifier	0.6144	0.6311	0.6144	0.6158	0.6133	0.2291	0.2302	0.2040
Decision Tree Classifier	0.5979	0.5983	0.5979	0.6011	0.5960	0.1964	0.1991	0.0810
Quadratic Discriminant Analysis	0.5588	0.5745	0.5588	0.5605	0.5563	0.1178	0.1193	0.0340
K Neighbors Classifier	0.5522	0.5630	0.5522	0.5531	0.5508	0.1046	0.1053	0.0490
SVM - Linear Kernel	0.5490	0.6459	0.5490	0.5906	0.4489	0.0972	0.1503	0.0590
Dummy Classifier	0.4934	0.5000	0.4934	0.2435	0.3261	0.0000	0.0000	0.0300

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 17: Resultados do modelo – Faixa 15

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Logistic Regression	0.6405	0.6782	0.6405	0.6438	0.6393	0.2815	0.2845	0.1860
Random Forest Classifier	0.6356	0.6810	0.6356	0.6384	0.6337	0.2714	0.2740	0.3450
Light Gradient Boosting Machine	0.6323	0.6837	0.6323	0.6352	0.6302	0.2642	0.2672	0.7640
Ridge Classifier	0.6290	0.6774	0.6290	0.6319	0.6277	0.2587	0.2611	0.0560
Linear Discriminant Analysis	0.6274	0.6772	0.6274	0.6303	0.6261	0.2555	0.2579	0.0320
Gradient Boosting Classifier	0.6258	0.6571	0.6258	0.6274	0.6245	0.2519	0.2533	0.7600
Extra Trees Classifier	0.6258	0.6877	0.6258	0.6303	0.6236	0.2519	0.2561	0.2210
Naive Bayes	0.6257	0.6609	0.6257	0.6287	0.6243	0.2514	0.2542	0.0470
Extreme Gradient Boosting	0.6208	0.6738	0.6208	0.6226	0.6190	0.2413	0.2432	0.2790
Ada Boost Classifier	0.5834	0.6203	0.5834	0.5842	0.5811	0.1668	0.1675	0.2090
Quadratic Discriminant Analysis	0.5801	0.5967	0.5801	0.5822	0.5779	0.1607	0.1625	0.0350
Decision Tree Classifier	0.5554	0.5551	0.5554	0.5557	0.5542	0.1101	0.1107	0.0760
K Neighbors Classifier	0.5425	0.5441	0.5425	0.5433	0.5406	0.0845	0.0854	0.0340
SVM - Linear Kernel	0.5391	0.6537	0.5391	0.5223	0.4227	0.0803	0.1118	0.0650
Dummy Classifier	0.4934	0.5000	0.4934	0.2435	0.3261	0.0000	0.0000	0.0280

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 18: Resultados do modelo – Faixa 16

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Extra Trees Classifier	0.6453	0.6874	0.6453	0.6468	0.6437	0.2901	0.2917	0.2160
Light Gradient Boosting Machine	0.6436	0.6735	0.6436	0.6464	0.6415	0.2869	0.2897	0.3040
Ridge Classifier	0.6404	0.6962	0.6404	0.6453	0.6379	0.2809	0.2855	0.0600
Linear Discriminant Analysis	0.6404	0.6958	0.6404	0.6453	0.6379	0.2809	0.2855	0.0320
Random Forest Classifier	0.6403	0.6942	0.6403	0.6413	0.6396	0.2802	0.2813	0.3520
Logistic Regression	0.6388	0.6980	0.6388	0.6430	0.6363	0.2776	0.2816	0.1990
Naive Bayes	0.6306	0.6785	0.6306	0.6325	0.6297	0.2613	0.2631	0.0570
Gradient Boosting Classifier	0.6306	0.6812	0.6306	0.6317	0.6295	0.2607	0.2620	0.7360
Extreme Gradient Boosting	0.6257	0.6630	0.6257	0.6280	0.6235	0.2513	0.2535	0.2740
Quadratic Discriminant Analysis	0.6141	0.6511	0.6141	0.6240	0.6072	0.2285	0.2379	0.0340
Ada Boost Classifier	0.5996	0.6397	0.5996	0.6025	0.5971	0.1995	0.2021	0.2060
Decision Tree Classifier	0.5702	0.5705	0.5702	0.5717	0.5684	0.1409	0.1420	0.0790
SVM - Linear Kernel	0.5655	0.6949	0.5655	0.5698	0.4902	0.1279	0.1637	0.0700
K Neighbors Classifier	0.5425	0.5678	0.5425	0.5427	0.5398	0.0846	0.0849	0.0350
Dummy Classifier	0.4934	0.5000	0.4934	0.2435	0.3261	0.0000	0.0000	0.0530

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews; F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

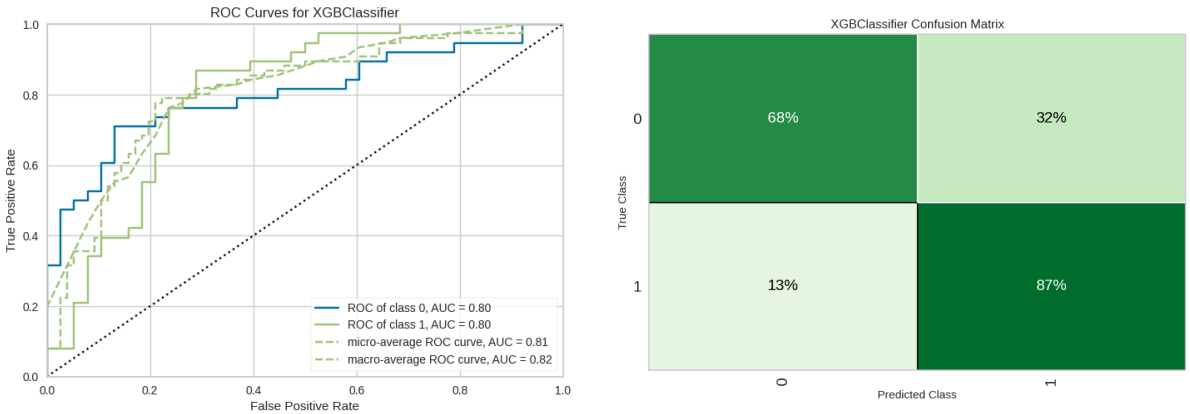

Figura 2: Matriz de Confusão e Curva ROC para o Modelo Faixa 1. Patologia: 0, Saudável: 1. Fonte: Gerado com o PyCaret (2025).

Considerando a acurácia como métrica principal de avaliação, a faixa 2, que foi treinada utilizando apenas os *MFCCs*, obteve o melhor desempenho, com valor de 0,7482. Ao analisar a coluna AUC, a faixa 2 também apresentou o melhor desempenho, alcançando 0,8313, o que indica uma melhor capacidade de discriminação entre as classes.

Em relação à matriz de confusão, a faixa 1 obteve o maior número de classificações corretas, especialmente nos casos de predições como classe saudável, o que resulta em um elevado índice de verdadeiros positivos. Esse desempenho pode ser atribuído à utilização das características acústicas no treinamento, que forneceram uma representação mais robusta e informativa dos dados.

Por outro lado, os modelos treinados com as faixas de frequência mais altas (a partir da faixa 6) apresentaram desempenho inferior, com acurácia abaixo de 0,71. Essa queda de desempenho pode ser atribuída à menor quantidade de informações relevantes presentes nesses intervalos. Os respectivos modelos apresentaram valores de AUC inferiores a 0,75, indicando uma maior dificuldade em distinguir corretamente entre as classes e refletindo uma perda significativa de desempenho em comparação às faixas de frequência mais baixas, nas quais o sinal útil era mais evidente.

O meta-modelo treinado neste cenário alcançou uma acurácia de 0,7800. Para efeito de comparação, a abordagem baseada exclusivamente na votação majoritária dos modelos individuais obteve uma acurácia de 0,7051, evidenciando a superioridade do meta-modelo. A seguir, são apresentadas a tabela de resultados e os gráficos correspondentes a esse modelo.

Tabela 19: Resultados de Desempenho do Meta-modelo

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Ada Boost Classifier	0.7800	0.8056	0.7800	0.7835	0.7593	0.5322	0.5599	0.3290
Ridge Classifier	0.7767	0.7944	0.7767	0.7867	0.7667	0.5297	0.5452	0.1330
K Neighbors Classifier	0.7600	0.8139	0.7600	0.7975	0.7539	0.5179	0.5488	0.2380
Naive Bayes	0.7600	0.8278	0.7600	0.8092	0.7520	0.5174	0.5600	0.2020
Light Gradient Boosting Machine	0.7533	0.7389	0.7533	0.7792	0.7405	0.4916	0.5153	0.3510
Logistic Regression	0.7400	0.7889	0.7400	0.7408	0.7324	0.4585	0.4632	0.1440
Linear Discriminant Analysis	0.7200	0.7444	0.7200	0.7300	0.7105	0.4189	0.4316	0.1300
Decision Tree Classifier	0.7167	0.6833	0.7167	0.7283	0.7086	0.4277	0.4339	0.1420
SVM - Linear Kernel	0.7000	0.7389	0.7000	0.7243	0.6706	0.3725	0.4121	0.1390
Extreme Gradient Boosting	0.7000	0.7611	0.7000	0.7342	0.6910	0.3877	0.4173	0.1630
Gradient Boosting Classifier	0.6967	0.7222	0.6967	0.7242	0.6962	0.4103	0.4207	0.2160
Random Forest Classifier	0.6833	0.8056	0.6833	0.7292	0.6677	0.3532	0.3933	0.2950
Extra Trees Classifier	0.6800	0.7667	0.6800	0.7117	0.6700	0.3486	0.3748	0.2580
Quadratic Discriminant Analysis	0.6433	0.7611	0.6433	0.6188	0.5904	0.2414	0.2729	0.1990
Dummy Classifier	0.4400	0.5000	0.4400	0.1960	0.2705	0.0000	0.0000	0.1280

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews; F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

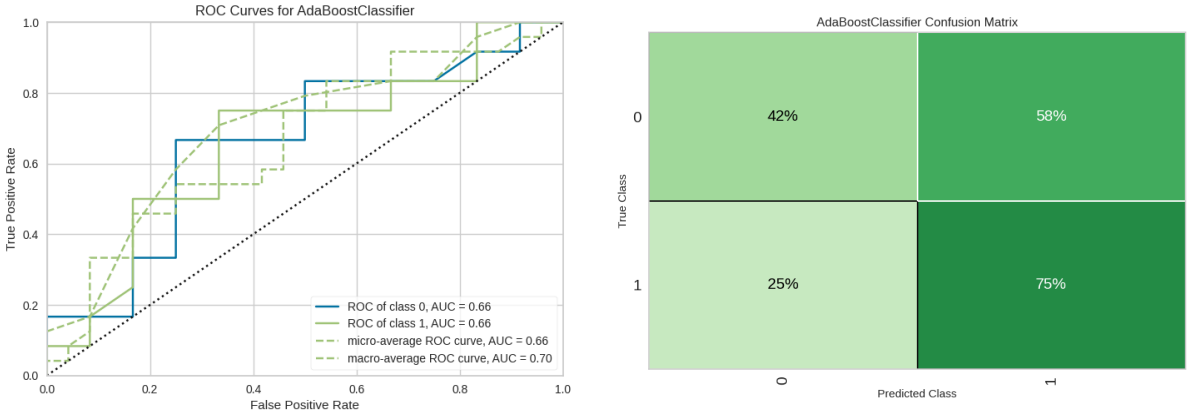

Figura 3: Matriz de Confusão e Curva ROC para o Meta Modelo. Patologia: 0, Saudável: 1. Fonte: Gerado com o PyCaret (2025).

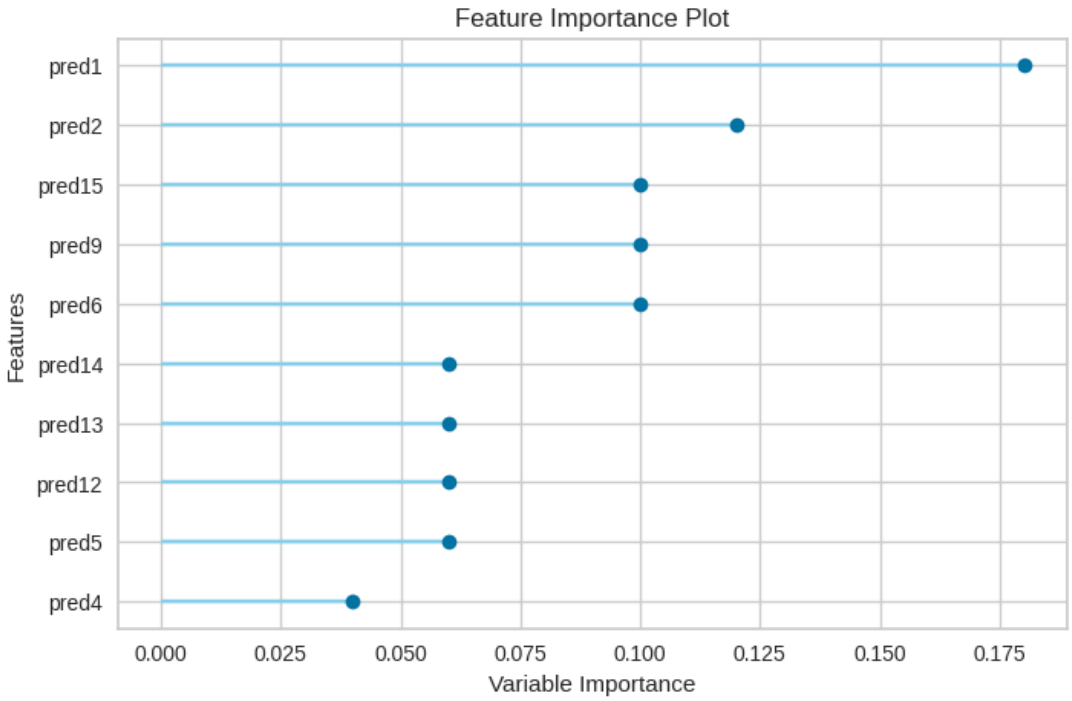

Figura 4: Gráfico de importância dos modelos individuais para a classificação final do meta modelo. O eixo y (*features*) representa o modelo da faixa correspondente. Fonte: Elaborado pelo autor com base em resultados gerados pelo PyCaret (2025).

5.2 Resultados Com Data Agumentation

A seguir, são apresentados os resultados dos modelos com aplicação de *data augmentation*.

Tabela 20: Resultados do Modelo com *Data Augmentation* – Faixa 1

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Extra Trees Classifier	0.8916	0.9634	0.8916	0.8931	0.8915	0.7832	0.7847	0.3810
Light Gradient Boosting Machine	0.8785	0.9464	0.8785	0.8801	0.8784	0.7570	0.7586	4.2870
Extreme Gradient Boosting	0.8780	0.9408	0.8780	0.8795	0.8778	0.7559	0.7574	0.8410
Random Forest Classifier	0.8763	0.9461	0.8763	0.8782	0.8762	0.7527	0.7545	0.8640
Gradient Boosting Classifier	0.8082	0.8964	0.8082	0.8110	0.8078	0.6164	0.6192	2.9380
Decision Tree Classifier	0.8077	0.8078	0.8077	0.8088	0.8076	0.6155	0.6165	0.1640
Ada Boost Classifier	0.7592	0.8417	0.7592	0.7627	0.7585	0.5184	0.5219	0.7740
Linear Discriminant Analysis	0.7151	0.8098	0.7151	0.7183	0.7141	0.4302	0.4334	0.0390
Ridge Classifier	0.7113	0.8048	0.7113	0.7136	0.7105	0.4225	0.4248	0.0800
Logistic Regression	0.7059	0.7755	0.7059	0.7073	0.7053	0.4117	0.4131	1.6430
Quadratic Discriminant Analysis	0.6901	0.7948	0.6901	0.7546	0.6678	0.3801	0.4395	0.0450
Naive Bayes	0.6580	0.7751	0.6580	0.7283	0.6278	0.3158	0.3791	0.0380
K Neighbors Classifier	0.5975	0.6511	0.5975	0.5993	0.5952	0.1949	0.1968	0.0690
SVM - Linear Kernel	0.5256	0.6777	0.5256	0.6914	0.4006	0.0506	0.1188	0.1130
Dummy Classifier	0.4989	0.5000	0.4989	0.2489	0.3321	0.0000	0.0000	0.0330

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 21: Resultados do Modelo com *Data Augmentation* – Faixa 2

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Extra Trees Classifier	0.8229	0.9110	0.8229	0.8239	0.8228	0.6459	0.6468	0.3790
Extreme Gradient Boosting	0.8088	0.8857	0.8088	0.8108	0.8085	0.6175	0.6196	0.4650
Random Forest Classifier	0.8055	0.8841	0.8055	0.8075	0.8051	0.6109	0.6130	0.8180
Light Gradient Boosting Machine	0.7989	0.8880	0.7989	0.8003	0.7986	0.5978	0.5992	2.6820
Gradient Boosting Classifier	0.7548	0.8372	0.7548	0.7566	0.7544	0.5096	0.5113	1.7170
Ada Boost Classifier	0.7140	0.7925	0.7140	0.7151	0.7137	0.4280	0.4290	0.5590
Ridge Classifier	0.7118	0.7888	0.7118	0.7150	0.7109	0.4236	0.4267	0.0650
Linear Discriminant Analysis	0.7118	0.7888	0.7118	0.7150	0.7109	0.4236	0.4267	0.0380
Logistic Regression	0.7113	0.7880	0.7113	0.7140	0.7105	0.4225	0.4252	0.1700
Naive Bayes	0.7058	0.7732	0.7058	0.7087	0.7046	0.4115	0.4144	0.0370
Decision Tree Classifier	0.6966	0.6966	0.6966	0.6975	0.6963	0.3932	0.3941	0.1240
K Neighbors Classifier	0.6858	0.7561	0.6858	0.6909	0.6836	0.3716	0.3766	0.0510
Quadratic Discriminant Analysis	0.6857	0.7851	0.6857	0.6963	0.6807	0.3712	0.3817	0.0390
SVM - Linear Kernel	0.6159	0.6966	0.6159	0.6641	0.5790	0.2319	0.2675	0.0920
Dummy Classifier	0.4989	0.5000	0.4989	0.2489	0.3321	0.0000	0.0000	0.0370

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 22: Resultados do Modelo com *Data Augmentation* – Faixa 3

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Extra Trees Classifier	0.7543	0.8405	0.7543	0.7560	0.7539	0.5086	0.5103	0.4140
Extreme Gradient Boosting	0.7413	0.8267	0.7413	0.7425	0.7409	0.4824	0.4837	0.4920
Light Gradient Boosting Machine	0.7391	0.8192	0.7391	0.7405	0.7387	0.4781	0.4795	3.5500
Random Forest Classifier	0.7314	0.8154	0.7314	0.7328	0.7310	0.4628	0.4642	1.0320
Gradient Boosting Classifier	0.7216	0.7964	0.7216	0.7222	0.7215	0.4433	0.4438	1.7700
Ada Boost Classifier	0.6786	0.7489	0.6786	0.6800	0.6779	0.3571	0.3586	0.4140
Ridge Classifier	0.6721	0.7454	0.6721	0.6739	0.6713	0.3441	0.3459	0.0410
Linear Discriminant Analysis	0.6721	0.7454	0.6721	0.6739	0.6713	0.3441	0.3459	0.0670
Naive Bayes	0.6715	0.7522	0.6715	0.6875	0.6637	0.3429	0.3584	0.0360
Logistic Regression	0.6704	0.7450	0.6704	0.6718	0.6699	0.3408	0.3422	0.2350
K Neighbors Classifier	0.6645	0.7106	0.6645	0.6662	0.6637	0.3289	0.3307	0.0490
Quadratic Discriminant Analysis	0.6459	0.7442	0.6459	0.6737	0.6295	0.2917	0.3181	0.0390
Decision Tree Classifier	0.6438	0.6438	0.6438	0.6453	0.6430	0.2876	0.2890	0.0800
SVM - Linear Kernel	0.5506	0.6789	0.5506	0.6002	0.4743	0.1004	0.1372	0.0520
Dummy Classifier	0.4989	0.5000	0.4989	0.2489	0.3321	0.0000	0.0000	0.0350

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 23: Resultados do Modelo com *Data Augmentation* – Faixa 4

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Extra Trees Classifier	0.7402	0.8242	0.7402	0.7416	0.7396	0.4803	0.4817	0.3650
Extreme Gradient Boosting	0.7391	0.8083	0.7391	0.7411	0.7384	0.4781	0.4801	0.7130
Random Forest Classifier	0.7336	0.8107	0.7336	0.7346	0.7333	0.4672	0.4682	0.7940
Light Gradient Boosting Machine	0.7325	0.8116	0.7325	0.7341	0.7320	0.4650	0.4666	2.8480
Gradient Boosting Classifier	0.7075	0.7800	0.7075	0.7088	0.7070	0.4149	0.4163	1.7390
Ada Boost Classifier	0.6791	0.7459	0.6791	0.6808	0.6781	0.3582	0.3599	0.6160
Ridge Classifier	0.6770	0.7428	0.6770	0.6784	0.6764	0.3539	0.3553	0.0840
Linear Discriminant Analysis	0.6770	0.7428	0.6770	0.6784	0.6764	0.3539	0.3553	0.0420
Logistic Regression	0.6715	0.7436	0.6715	0.6726	0.6710	0.3430	0.3441	0.2260
Naive Bayes	0.6655	0.7484	0.6655	0.6909	0.6534	0.3310	0.3552	0.0660
K Neighbors Classifier	0.6438	0.6989	0.6438	0.6442	0.6435	0.2875	0.2880	0.0600
Quadratic Discriminant Analysis	0.6421	0.7269	0.6421	0.6768	0.6222	0.2841	0.3166	0.0400
Decision Tree Classifier	0.6362	0.6362	0.6362	0.6374	0.6354	0.2724	0.2736	0.1220
SVM - Linear Kernel	0.5207	0.6542	0.5207	0.5472	0.4073	0.0430	0.0680	0.0890
Dummy Classifier	0.4989	0.5000	0.4989	0.2489	0.3321	0.0000	0.0000	0.0340

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 24: Resultados do Modelo com *Data Augmentation* – Faixa 5

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Extra Trees Classifier	0.7298	0.8004	0.7298	0.7318	0.7292	0.4595	0.4616	0.3660
Extreme Gradient Boosting	0.7151	0.7878	0.7151	0.7171	0.7144	0.4301	0.4321	0.7270
Light Gradient Boosting Machine	0.7129	0.7999	0.7129	0.7144	0.7122	0.4257	0.4273	3.2850
Random Forest Classifier	0.7108	0.7842	0.7108	0.7124	0.7101	0.4214	0.4231	0.8410
Gradient Boosting Classifier	0.6988	0.7794	0.6988	0.7012	0.6977	0.3974	0.3999	1.7170
Ada Boost Classifier	0.6906	0.7504	0.6906	0.6921	0.6900	0.3812	0.3827	0.6290
Ridge Classifier	0.6873	0.7426	0.6873	0.6880	0.6871	0.3747	0.3754	0.0630
Linear Discriminant Analysis	0.6873	0.7427	0.6873	0.6880	0.6871	0.3747	0.3754	0.0410
Logistic Regression	0.6835	0.7427	0.6835	0.6845	0.6832	0.3671	0.3680	0.2450
Naive Bayes	0.6486	0.7372	0.6486	0.6791	0.6315	0.2972	0.3260	0.0420
Decision Tree Classifier	0.6269	0.6268	0.6269	0.6277	0.6262	0.2537	0.2545	0.1400
Quadratic Discriminant Analysis	0.6219	0.7255	0.6219	0.6601	0.5964	0.2438	0.2788	0.0420
K Neighbors Classifier	0.6198	0.6621	0.6198	0.6208	0.6190	0.2395	0.2406	0.0500
SVM - Linear Kernel	0.5647	0.6754	0.5647	0.6036	0.4859	0.1286	0.1573	0.0970
Dummy Classifier	0.4989	0.5000	0.4989	0.2489	0.3321	0.0000	0.0000	0.0360

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 25: Resultados do Modelo com *Data Augmentation* – Faixa 6

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Light Gradient Boosting Machine	0.6879	0.7496	0.6879	0.6886	0.6876	0.3757	0.3764	2.1940
Extreme Gradient Boosting	0.6863	0.7429	0.6863	0.6867	0.6861	0.3725	0.3729	0.4820
Extra Trees Classifier	0.6846	0.7577	0.6846	0.6860	0.6839	0.3691	0.3705	0.5980
Random Forest Classifier	0.6824	0.7405	0.6824	0.6839	0.6818	0.3648	0.3663	1.0480
Gradient Boosting Classifier	0.6748	0.7371	0.6748	0.6757	0.6743	0.3495	0.3504	1.7330
Ridge Classifier	0.6530	0.7014	0.6530	0.6538	0.6525	0.3059	0.3068	0.0790
Linear Discriminant Analysis	0.6530	0.7014	0.6530	0.6538	0.6525	0.3059	0.3068	0.0420
Logistic Regression	0.6514	0.7020	0.6514	0.6521	0.6508	0.3026	0.3034	0.3780
Ada Boost Classifier	0.6356	0.6864	0.6356	0.6369	0.6347	0.2712	0.2724	0.3900
Naive Bayes	0.6007	0.6966	0.6007	0.6316	0.5749	0.2014	0.2300	0.0400
Quadratic Discriminant Analysis	0.6002	0.6895	0.6002	0.6376	0.5697	0.2003	0.2346	0.0470
K Neighbors Classifier	0.5872	0.6241	0.5872	0.5880	0.5862	0.1742	0.1751	0.0900
Decision Tree Classifier	0.5839	0.5838	0.5839	0.5848	0.5829	0.1677	0.1686	0.0830
SVM - Linear Kernel	0.5256	0.6609	0.5256	0.6024	0.4065	0.0516	0.0960	0.0560
Dummy Classifier	0.4989	0.5000	0.4989	0.2489	0.3321	0.0000	0.0000	0.0620

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 26: Resultados do Modelo com *Data Augmentation* – Faixa 7

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Light Gradient Boosting Machine	0.6443	0.7013	0.6443	0.6449	0.6438	0.2884	0.2891	2.7280
Extra Trees Classifier	0.6416	0.7046	0.6416	0.6423	0.6411	0.2831	0.2838	0.4180
Random Forest Classifier	0.6400	0.7003	0.6400	0.6409	0.6392	0.2798	0.2808	0.9140
Extreme Gradient Boosting	0.6389	0.6919	0.6389	0.6397	0.6383	0.2776	0.2785	0.6810
Gradient Boosting Classifier	0.6361	0.6926	0.6361	0.6368	0.6355	0.2721	0.2729	1.7280
Logistic Regression	0.6067	0.6513	0.6067	0.6077	0.6053	0.2132	0.2143	0.2400
Ridge Classifier	0.6051	0.6493	0.6051	0.6063	0.6033	0.2100	0.2112	0.0690
Linear Discriminant Analysis	0.6051	0.6493	0.6051	0.6063	0.6033	0.2100	0.2112	0.0440
Ada Boost Classifier	0.6002	0.6416	0.6002	0.6009	0.5992	0.2002	0.2010	0.6000
Quadratic Discriminant Analysis	0.5931	0.6627	0.5931	0.6317	0.5589	0.1861	0.2212	0.0470
Naive Bayes	0.5925	0.6498	0.5925	0.6247	0.5633	0.1849	0.2145	0.0430
Decision Tree Classifier	0.5594	0.5593	0.5594	0.5597	0.5586	0.1187	0.1190	0.0970
K Neighbors Classifier	0.5534	0.5738	0.5534	0.5539	0.5524	0.1067	0.1072	0.0570
SVM - Linear Kernel	0.5163	0.6236	0.5163	0.5796	0.3816	0.0325	0.0786	0.1420
Dummy Classifier	0.4989	0.5000	0.4989	0.2489	0.3321	0.0000	0.0000	0.0620

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 27: Resultados do Modelo com *Data Augmentation* – Faixa 8

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Random Forest Classifier	0.6324	0.6845	0.6324	0.6334	0.6315	0.2646	0.2657	0.9350
Extra Trees Classifier	0.6286	0.6901	0.6286	0.6297	0.6276	0.2570	0.2582	0.3810
Light Gradient Boosting Machine	0.6231	0.6763	0.6231	0.6237	0.6223	0.2461	0.2467	2.8320
Gradient Boosting Classifier	0.6133	0.6638	0.6133	0.6138	0.6127	0.2265	0.2271	1.7630
Extreme Gradient Boosting	0.6095	0.6662	0.6095	0.6097	0.6092	0.2190	0.2192	0.6220
Logistic Regression	0.6051	0.6474	0.6051	0.6058	0.6043	0.2102	0.2109	0.2410
Ada Boost Classifier	0.6040	0.6620	0.6040	0.6048	0.6032	0.2080	0.2088	0.5340
Ridge Classifier	0.6002	0.6441	0.6002	0.6012	0.5991	0.2004	0.2014	0.0710
Linear Discriminant Analysis	0.6002	0.6441	0.6002	0.6012	0.5991	0.2004	0.2014	0.0460
Quadratic Discriminant Analysis	0.5850	0.6311	0.5850	0.6031	0.5660	0.1699	0.1870	0.0470
Naive Bayes	0.5746	0.6520	0.5746	0.5872	0.5578	0.1490	0.1611	0.0460
K Neighbors Classifier	0.5621	0.5740	0.5621	0.5626	0.5613	0.1240	0.1246	0.0530
Decision Tree Classifier	0.5446	0.5446	0.5446	0.5448	0.5441	0.0893	0.0895	0.0870
SVM - Linear Kernel	0.5316	0.6211	0.5316	0.6370	0.4173	0.0624	0.1284	0.0970
Dummy Classifier	0.4989	0.5000	0.4989	0.2489	0.3321	0.0000	0.0000	0.0600

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 28: Resultados do Modelo com *Data Augmentation* – Faixa 9

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Random Forest Classifier	0.6481	0.6872	0.6481	0.6494	0.6473	0.2961	0.2974	0.9250
Gradient Boosting Classifier	0.6389	0.6935	0.6389	0.6399	0.6381	0.2776	0.2788	1.7400
Extra Trees Classifier	0.6367	0.6812	0.6367	0.6372	0.6363	0.2733	0.2739	0.3800
Light Gradient Boosting Machine	0.6280	0.6752	0.6280	0.6289	0.6272	0.2558	0.2568	3.4910
Extreme Gradient Boosting	0.6253	0.6736	0.6253	0.6271	0.6239	0.2503	0.2523	0.6580
Logistic Regression	0.6133	0.6661	0.6133	0.6140	0.6126	0.2264	0.2272	0.2800
Naive Bayes	0.6127	0.6686	0.6127	0.6177	0.6066	0.2252	0.2302	0.0460
Ridge Classifier	0.6111	0.6633	0.6111	0.6119	0.6104	0.2221	0.2229	0.0680
Linear Discriminant Analysis	0.6111	0.6633	0.6111	0.6119	0.6104	0.2221	0.2229	0.0440
Ada Boost Classifier	0.6024	0.6431	0.6024	0.6032	0.6015	0.2046	0.2055	0.5930
Quadratic Discriminant Analysis	0.6019	0.6517	0.6019	0.6038	0.5977	0.2036	0.2056	0.0470
Decision Tree Classifier	0.5534	0.5534	0.5534	0.5535	0.5532	0.1068	0.1069	0.1110
K Neighbors Classifier	0.5338	0.5334	0.5338	0.5341	0.5330	0.0675	0.0678	0.0590
SVM - Linear Kernel	0.5316	0.6623	0.5316	0.5738	0.4315	0.0639	0.1051	0.0970
Dummy Classifier	0.4989	0.5000	0.4989	0.2489	0.3321	0.0000	0.0000	0.0440

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 29: Resultados do Modelo com *Data Augmentation* – Faixa 10

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Gradient Boosting Classifier	0.6383	0.6845	0.6383	0.6390	0.6378	0.2766	0.2773	1.7320
Extra Trees Classifier	0.6351	0.6954	0.6351	0.6354	0.6348	0.2701	0.2705	0.6070
Random Forest Classifier	0.6290	0.6888	0.6290	0.6297	0.6285	0.2580	0.2587	1.0650
Naive Bayes	0.6241	0.6845	0.6241	0.6260	0.6225	0.2480	0.2500	0.0470
Quadratic Discriminant Analysis	0.6214	0.6609	0.6214	0.6232	0.6197	0.2426	0.2445	0.0460
Logistic Regression	0.6198	0.6774	0.6198	0.6204	0.6194	0.2395	0.2402	0.3840
Extreme Gradient Boosting	0.6155	0.6657	0.6155	0.6159	0.6151	0.2309	0.2313	0.5350
Light Gradient Boosting Machine	0.6154	0.6637	0.6154	0.6162	0.6146	0.2308	0.2316	2.7800
Ridge Classifier	0.6143	0.6746	0.6143	0.6150	0.6139	0.2287	0.2293	0.0490
Linear Discriminant Analysis	0.6143	0.6746	0.6143	0.6150	0.6139	0.2287	0.2293	0.0490
Ada Boost Classifier	0.6056	0.6521	0.6056	0.6063	0.6051	0.2112	0.2119	0.4270
Decision Tree Classifier	0.5534	0.5533	0.5534	0.5535	0.5530	0.1067	0.1068	0.0920
K Neighbors Classifier	0.5463	0.5670	0.5463	0.5469	0.5443	0.0926	0.0932	0.0850
SVM - Linear Kernel	0.5316	0.6582	0.5316	0.6012	0.4243	0.0640	0.1042	0.0660
Dummy Classifier	0.4989	0.5000	0.4989	0.2489	0.3321	0.0000	0.0000	0.0460

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 30: Resultados do Modelo com *Data Augmentation* – Faixa 11

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Extra Trees Classifier	0.6378	0.6927	0.6378	0.6386	0.6372	0.2755	0.2763	0.3860
Random Forest Classifier	0.6361	0.6808	0.6361	0.6369	0.6356	0.2722	0.2730	1.0730
Gradient Boosting Classifier	0.6302	0.6871	0.6302	0.6313	0.6294	0.2603	0.2615	1.8600
Naive Bayes	0.6225	0.6758	0.6225	0.6242	0.6210	0.2450	0.2467	0.0510
Logistic Regression	0.6209	0.6741	0.6209	0.6216	0.6202	0.2418	0.2425	0.3150
Ridge Classifier	0.6198	0.6735	0.6198	0.6209	0.6188	0.2396	0.2407	0.0520
Linear Discriminant Analysis	0.6198	0.6734	0.6198	0.6209	0.6188	0.2396	0.2407	0.0730
Extreme Gradient Boosting	0.6193	0.6833	0.6193	0.6198	0.6188	0.2385	0.2390	0.5420
Light Gradient Boosting Machine	0.6165	0.6860	0.6165	0.6168	0.6163	0.2330	0.2333	2.3710
Quadratic Discriminant Analysis	0.6160	0.6599	0.6160	0.6168	0.6151	0.2320	0.2328	0.0460
Ada Boost Classifier	0.5997	0.6588	0.5997	0.6007	0.5989	0.1994	0.2003	0.4570
SVM - Linear Kernel	0.5572	0.6587	0.5572	0.6487	0.4725	0.1157	0.1730	0.0680
Decision Tree Classifier	0.5458	0.5457	0.5458	0.5460	0.5450	0.0914	0.0917	0.0910
K Neighbors Classifier	0.5305	0.5438	0.5305	0.5307	0.5288	0.0610	0.0612	0.0720
Dummy Classifier	0.4989	0.5000	0.4989	0.2489	0.3321	0.0000	0.0000	0.0420

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 31: Resultados do Modelo com *Data Augmentation* – Faixa 12

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Extra Trees Classifier	0.6209	0.6692	0.6209	0.6214	0.6204	0.2417	0.2423	0.4570
Random Forest Classifier	0.6160	0.6698	0.6160	0.6168	0.6152	0.2318	0.2327	1.0560
Naive Bayes	0.6116	0.6505	0.6116	0.6132	0.6099	0.2232	0.2248	0.0510
Extreme Gradient Boosting	0.5991	0.6365	0.5991	0.6009	0.5975	0.1981	0.1999	0.4940
Ridge Classifier	0.5964	0.6420	0.5964	0.5979	0.5947	0.1927	0.1943	0.0520
Linear Discriminant Analysis	0.5958	0.6420	0.5958	0.5973	0.5942	0.1916	0.1931	0.0740
Light Gradient Boosting Machine	0.5958	0.6374	0.5958	0.5964	0.5950	0.1914	0.1921	2.2100
Logistic Regression	0.5948	0.6427	0.5948	0.5959	0.5936	0.1895	0.1906	0.3880
Gradient Boosting Classifier	0.5926	0.6444	0.5926	0.5933	0.5916	0.1851	0.1858	1.9830
Ada Boost Classifier	0.5877	0.6310	0.5877	0.5882	0.5871	0.1752	0.1758	0.4010
Quadratic Discriminant Analysis	0.5866	0.6137	0.5866	0.5873	0.5856	0.1731	0.1738	0.0510
Decision Tree Classifier	0.5615	0.5615	0.5615	0.5621	0.5606	0.1230	0.1235	0.0960
K Neighbors Classifier	0.5229	0.5198	0.5229	0.5231	0.5218	0.0456	0.0459	0.0620
SVM - Linear Kernel	0.5223	0.6170	0.5223	0.6119	0.4085	0.0455	0.0872	0.0690
Dummy Classifier	0.4989	0.5000	0.4989	0.2489	0.3321	0.0000	0.0000	0.0460

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 32: Resultados do Modelo com *Data Augmentation* – Faixa 13

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Naive Bayes	0.6258	0.6600	0.6258	0.6310	0.6215	0.2515	0.2566	0.0730
Extra Trees Classifier	0.6214	0.6650	0.6214	0.6227	0.6204	0.2427	0.2440	0.4190
Random Forest Classifier	0.6143	0.6686	0.6143	0.6153	0.6134	0.2285	0.2296	0.9430
Extreme Gradient Boosting	0.6100	0.6564	0.6100	0.6109	0.6090	0.2198	0.2208	0.6390
Light Gradient Boosting Machine	0.6018	0.6512	0.6018	0.6023	0.6012	0.2035	0.2040	2.1570
Ada Boost Classifier	0.5958	0.6360	0.5958	0.5962	0.5953	0.1915	0.1919	0.4300
Gradient Boosting Classifier	0.5926	0.6474	0.5926	0.5932	0.5917	0.1850	0.1857	1.6720
Quadratic Discriminant Analysis	0.5904	0.6255	0.5904	0.5924	0.5876	0.1806	0.1826	0.0750
Logistic Regression	0.5876	0.6376	0.5876	0.5889	0.5865	0.1751	0.1765	0.3330
Ridge Classifier	0.5855	0.6371	0.5855	0.5869	0.5841	0.1708	0.1723	0.0520
Linear Discriminant Analysis	0.5855	0.6370	0.5855	0.5869	0.5841	0.1708	0.1723	0.0530
Decision Tree Classifier	0.5599	0.5599	0.5599	0.5600	0.5596	0.1197	0.1199	0.1380
K Neighbors Classifier	0.5430	0.5483	0.5430	0.5433	0.5424	0.0860	0.0862	0.0860
SVM - Linear Kernel	0.5272	0.6115	0.5272	0.6370	0.4138	0.0552	0.1013	0.0720
Dummy Classifier	0.4989	0.5000	0.4989	0.2489	0.3321	0.0000	0.0000	0.0690

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 33: Resultados do Modelo com *Data Augmentation* – Faixa 14

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Naive Bayes	0.6133	0.6439	0.6133	0.6205	0.6074	0.2264	0.2336	0.0490
Gradient Boosting Classifier	0.6127	0.6442	0.6127	0.6150	0.6112	0.2254	0.2277	1.7430
Random Forest Classifier	0.6062	0.6443	0.6062	0.6077	0.6048	0.2123	0.2138	1.0500
Extra Trees Classifier	0.6035	0.6465	0.6035	0.6048	0.6021	0.2069	0.2083	0.3930
Logistic Regression	0.5866	0.6209	0.5866	0.5875	0.5854	0.1730	0.1740	0.2640
Ridge Classifier	0.5844	0.6202	0.5844	0.5856	0.5830	0.1686	0.1698	0.0540
Linear Discriminant Analysis	0.5844	0.6201	0.5844	0.5856	0.5830	0.1686	0.1698	0.0740
Ada Boost Classifier	0.5806	0.6223	0.5806	0.5814	0.5799	0.1612	0.1620	0.4360
Light Gradient Boosting Machine	0.5801	0.6128	0.5801	0.5808	0.5791	0.1601	0.1609	3.1380
Extreme Gradient Boosting	0.5768	0.6101	0.5768	0.5772	0.5763	0.1537	0.1540	0.5060
Quadratic Discriminant Analysis	0.5665	0.5911	0.5665	0.5699	0.5609	0.1329	0.1363	0.0540
SVM - Linear Kernel	0.5245	0.5999	0.5245	0.5487	0.4103	0.0487	0.0822	0.0670
K Neighbors Classifier	0.5153	0.5222	0.5153	0.5154	0.5142	0.0306	0.0307	0.0630
Decision Tree Classifier	0.5141	0.5141	0.5141	0.5142	0.5137	0.0283	0.0283	0.0980
Dummy Classifier	0.4989	0.5000	0.4989	0.2489	0.3321	0.0000	0.0000	0.0680

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews;
F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 34: Resultados do Modelo com *Data Augmentation* – Faixa 15

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Random Forest Classifier	0.6040	0.6400	0.6040	0.6050	0.6029	0.2078	0.2089	0.9590
Gradient Boosting Classifier	0.5991	0.6284	0.5991	0.5999	0.5982	0.1981	0.1989	1.7080
Extra Trees Classifier	0.5975	0.6329	0.5975	0.5980	0.5969	0.1949	0.1954	0.3950
Naive Bayes	0.5921	0.6225	0.5921	0.5963	0.5875	0.1840	0.1882	0.0540
Extreme Gradient Boosting	0.5806	0.6084	0.5806	0.5810	0.5798	0.1610	0.1615	0.6380
Logistic Regression	0.5713	0.6128	0.5713	0.5720	0.5704	0.1425	0.1433	0.2620
Quadratic Discriminant Analysis	0.5709	0.5998	0.5709	0.5748	0.5649	0.1417	0.1456	0.0530
Ridge Classifier	0.5680	0.6121	0.5680	0.5689	0.5670	0.1360	0.1369	0.0750
Linear Discriminant Analysis	0.5680	0.6121	0.5680	0.5689	0.5670	0.1360	0.1369	0.0550
Light Gradient Boosting Machine	0.5610	0.6018	0.5610	0.5614	0.5602	0.1218	0.1223	2.4940
Ada Boost Classifier	0.5561	0.5805	0.5561	0.5564	0.5555	0.1120	0.1124	0.5430
K Neighbors Classifier	0.5392	0.5426	0.5392	0.5396	0.5379	0.0782	0.0787	0.0630
Decision Tree Classifier	0.5376	0.5375	0.5376	0.5377	0.5369	0.0751	0.0753	0.0980
SVM - Linear Kernel	0.5311	0.6110	0.5311	0.5370	0.4211	0.0628	0.0756	0.1070
Dummy Classifier	0.4989	0.5000	0.4989	0.2489	0.3321	0.0000	0.0000	0.0630

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews; F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

Tabela 35: Resultados do Modelo com *Data Augmentation* – Faixa 16

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Random Forest Classifier	0.6138	0.6619	0.6138	0.6147	0.6128	0.2275	0.2284	0.9920
Logistic Regression	0.6127	0.6675	0.6127	0.6143	0.6113	0.2254	0.2269	0.3030
Ridge Classifier	0.6111	0.6686	0.6111	0.6131	0.6092	0.2221	0.2242	0.0770
Linear Discriminant Analysis	0.6111	0.6685	0.6111	0.6131	0.6092	0.2221	0.2242	0.0570
Extra Trees Classifier	0.6024	0.6543	0.6024	0.6031	0.6014	0.2046	0.2054	0.3980
Quadratic Discriminant Analysis	0.6023	0.6504	0.6023	0.6160	0.5894	0.2045	0.2178	0.0570
Gradient Boosting Classifier	0.5997	0.6507	0.5997	0.6003	0.5985	0.1993	0.1999	1.7410
Extreme Gradient Boosting	0.5909	0.6360	0.5909	0.5918	0.5898	0.1816	0.1826	0.5070
Naive Bayes	0.5877	0.6317	0.5877	0.5947	0.5784	0.1752	0.1821	0.0540
Light Gradient Boosting Machine	0.5876	0.6369	0.5876	0.5884	0.5867	0.1752	0.1760	3.3980
Ada Boost Classifier	0.5703	0.6197	0.5703	0.5706	0.5694	0.1404	0.1408	0.5180
K Neighbors Classifier	0.5637	0.5839	0.5637	0.5646	0.5623	0.1273	0.1282	0.0640
SVM - Linear Kernel	0.5583	0.6479	0.5583	0.6117	0.4720	0.1166	0.1582	0.0850
Decision Tree Classifier	0.5479	0.5478	0.5479	0.5481	0.5466	0.0956	0.0959	0.1000
Dummy Classifier	0.4989	0.5000	0.4989	0.2489	0.3321	0.0000	0.0000	0.0530

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews; F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

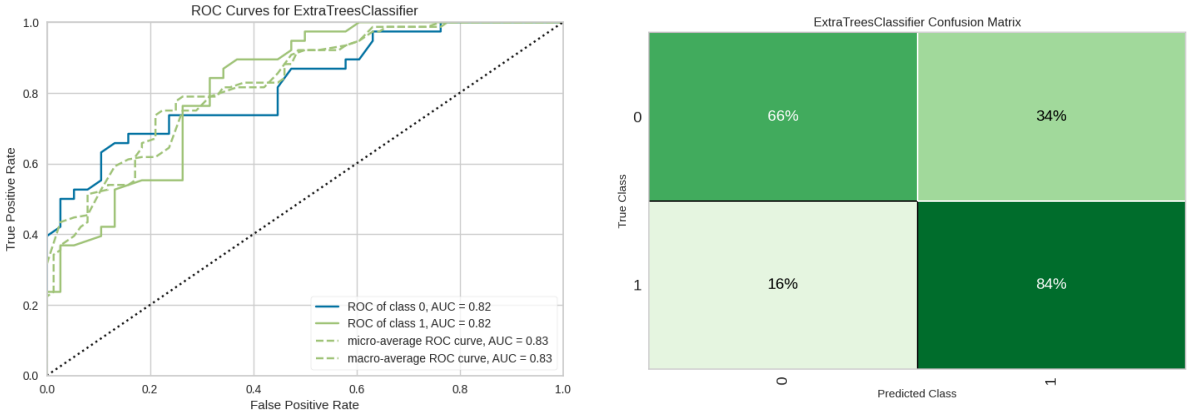

Figura 5: Matriz de Confusão e Curva ROC para o Modelo Faixa 1 com *Data Augmentation*. Patologia: 0, Saudável: 1. Fonte: Gerado com o PyCaret (2025).

A aplicação da técnica de *data augmentation* resultou em um aumento significativo na acurácia de alguns modelos. Na faixa 1, por exemplo, a acurácia passou de 0,7419 para 0,8916, representando um ganho de aproximadamente 20%. Na faixa 2, o desempenho aumentou de 0,7482 para 0,8229, um ganho de aproximadamente 10%. Embora, de maneira geral, a maioria dos modelos tenha apresentado melhorias, alguns sofreram queda de desempenho. Um exemplo disso ocorreu na faixa 15, cuja acurácia diminuiu de 0,6405 para 0,6040, uma redução de cerca de 5,7%.

No caso do meta-modelo, a ampliação da base de dados resultou em melhorias tanto na acurácia quanto na curva ROC. A acurácia aumentou de 0,7800 para 0,8233, enquanto a AUC da curva ROC, que antes era de 0,73, passou para 0,75. Além disso, houve uma redução significativa nos erros de classificação de dados saudáveis como patológicos. Inicialmente, o meta-modelo apresentava uma taxa de erro de 56% para essas previsões, mas com a aplicação de *data augmentation*, essa taxa caiu para 33%, evidenciando uma melhora considerável no desempenho do modelo. A influência do balanceamento realizado pelo meta-modelo foi significativa, visto que, sem ele, utilizando a votação majoritária dos modelos individuais, a acurácia é de 0,7564. Com o meta-modelo, no entanto, esse valor subiu para 0,8233. A seguir, são apresentados os resultados do meta-modelo.

Tabela 36: Resultados de Desempenho do Meta-modelo com *Data Augmentation*

Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Kappa	MCC	Tempo (s)
Light Gradient Boosting Machine	0.8233	0.8222	0.8233	0.8475	0.8200	0.6513	0.6707	0.2910
Naive Bayes	0.7600	0.8500	0.7600	0.7825	0.7562	0.5110	0.5319	0.1440
Logistic Regression	0.7500	0.8111	0.7500	0.7517	0.7393	0.4949	0.4967	0.1380
Random Forest Classifier	0.7467	0.7944	0.7467	0.7683	0.7429	0.4897	0.5081	0.3250
Ridge Classifier	0.7300	0.7944	0.7300	0.7367	0.7174	0.4494	0.4579	0.1360
Linear Discriminant Analysis	0.7300	0.7778	0.7300	0.7325	0.7193	0.4494	0.4539	0.1960
K Neighbors Classifier	0.7200	0.7500	0.7200	0.7458	0.7071	0.4250	0.4486	0.1350
Ada Boost Classifier	0.7133	0.7278	0.7133	0.7158	0.7012	0.4161	0.4206	0.2120
Gradient Boosting Classifier	0.7067	0.7278	0.7067	0.6983	0.6917	0.3918	0.3892	0.3750
Extra Trees Classifier	0.7067	0.7556	0.7067	0.7108	0.6964	0.4064	0.4093	0.2550
SVM - Linear Kernel	0.6800	0.6611	0.6800	0.6867	0.6650	0.3552	0.3611	0.2390
Extreme Gradient Boosting	0.6700	0.7889	0.6700	0.6817	0.6593	0.3410	0.3467	0.1630
Decision Tree Classifier	0.6333	0.6333	0.6333	0.6558	0.6170	0.2686	0.2819	0.2000
Quadratic Discriminant Analysis	0.5267	0.4222	0.5267	0.5102	0.4764	0.0541	0.0806	0.1330
Dummy Classifier	0.4400	0.5000	0.4400	0.1960	0.2705	0.0000	0.0000	0.2790

Fonte: Dados gerados pelo PyCaret (2025). Elaborado pelo autor.
Acurácia: acerto geral; AUC: área sob a curva ROC; MCC: coeficiente de correlação de Matthews; F1-Score: média harmônica entre precisão e revocação; Tempo (s): tempo de treinamento, em segundos.

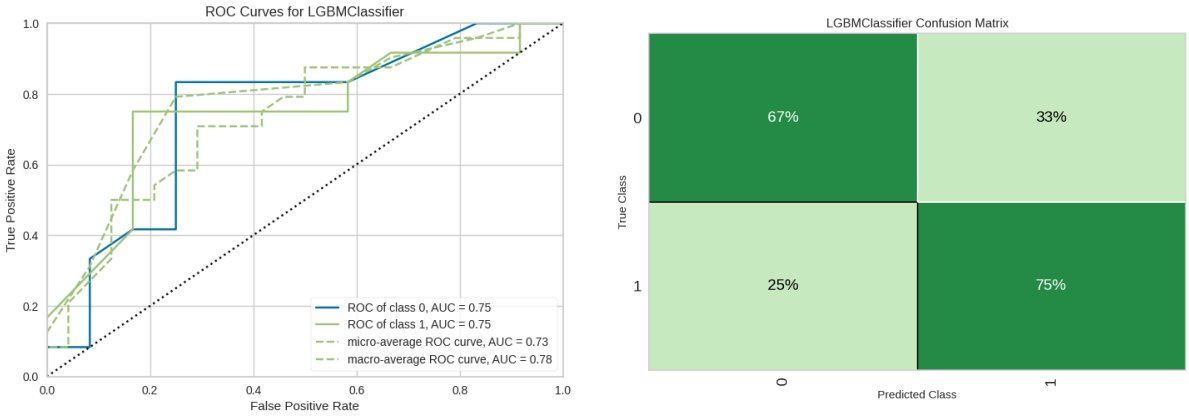

Figura 6: Matriz de Confusão e Curva ROC para o Meta Modelo com *Data Augmentation*. Patologia: 0, Saudável: 1. Fonte: Gerado com o PyCaret (2025).

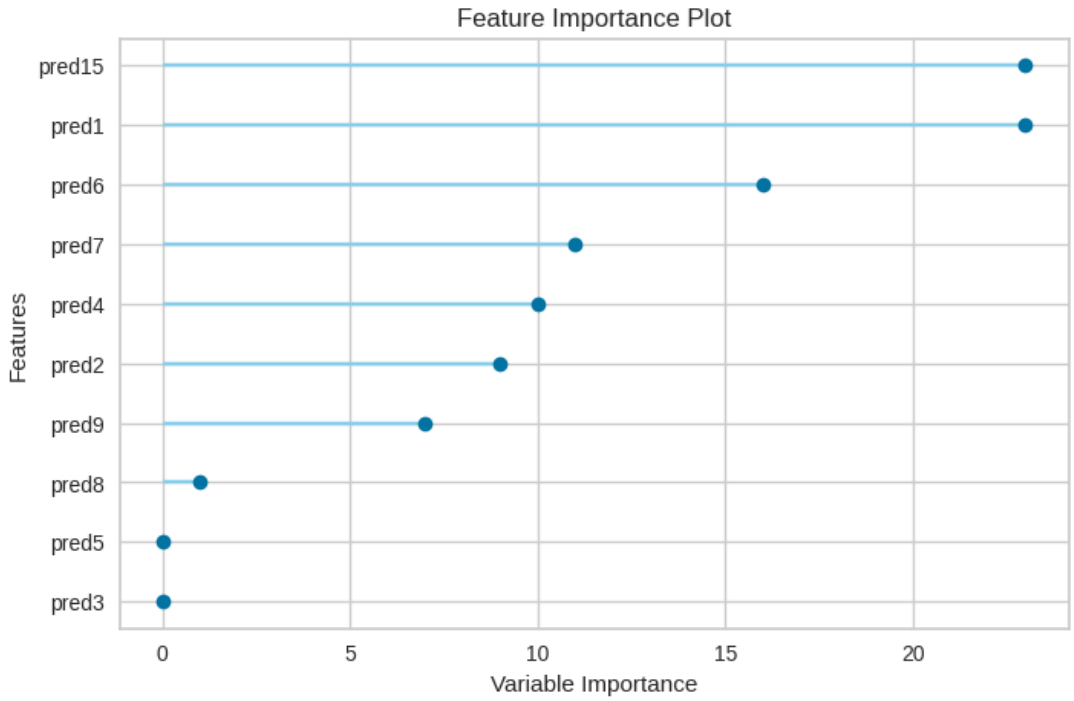

Figura 7: Gráfico de importância dos modelos individuais para a classificação final do meta modelo. O eixo y (*features*) representa o modelo da faixa correspondente. Fonte: Elaborado pelo autor com base em resultados gerados pelo PyCaret (2025).

Ao comparar os resultados com o artigo de (PAULINO et al., 2024), que obteve uma acurácia de 82,12% utilizando a mesma base de dados, o método proposto apresentou um desempenho ligeiramente superior, alcançando 82,33% no meta-modelo e uma acurácia máxima de 89,16% nos modelos individuais. Vale ressaltar que o resultado superior só foi alcançado com a aplicação de *data augmentation*; sem essa técnica, a acurácia do meta-modelo foi inferior, ficando em 78%. Esse resultado foi alcançado apesar do uso de métodos de aprendizado de máquina, que são computacionalmente menos custosos em comparação com as redes neurais convolucionais empregadas no artigo mencionado.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou uma abordagem baseada em aprendizado de máquina para a detecção de distúrbios vocais, explorando a extração de características acústicas e a influência das faixas de frequência na precisão dos modelos. A partir das técnicas de processamento de sinais, foram extraídas informações relevantes, como coeficientes *MFCC*, (*pitch*), *jitter*, *shimmer* e *formantes*, possibilitando a construção de um modelo de classificação para cada faixa de frequência.

Os experimentos demonstraram que a escolha das faixas de frequência impacta diretamente o desempenho dos classificadores, sendo que as faixas mais baixas apresentaram os melhores resultados, enquanto as faixas mais altas mostraram menor relevância para a detecção de distúrbios vocais.

Além disso, os resultados indicam que a abordagem proposta superou técnicas anteriores baseadas em *CNNs* e espectrogramas, evidenciando que métodos baseados em aprendizado de máquina tradicional podem ser eficazes para a tarefa de classificação de distúrbios vocais.

Por fim, este estudo contribui para o avanço das pesquisas na área de análise de voz ao evidenciar a importância da combinação entre características acústicas e coeficientes Mel. Como perspectivas futuras, sugere-se a exploração de novas abordagens para a extração de características nas faixas de frequência mais altas, utilizando espectrogramas. Além disso, o aumento da base de dados mostra-se essencial para aprimorar a acurácia do meta-modelo, uma vez que a Faixa 1 (modelo individual) obteve desempenho superior ao meta-modelo, possivelmente devido à limitação da quantidade de dados disponíveis para seu treinamento. Portanto, uma atenção especial à ampliação da base de dados pode ser determinante para ganhos de desempenho em estudos futuros.

REFERÊNCIAS

BOERSMA, P.; WEENINK, D. *Praat: doing phonetics by computer*. 2025. Acesso em: 31 mar. 2025. Disponível em: ⟨https://www.fon.hum.uva.nl/praat/⟩.

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

LOGAN, B. Mel frequency cepstral coefficients for music modeling. *International Symposium on Music Information Retrieval*, 2000.

MAGDIN, M. et al. Voice analysis using praat software and classification of user emotional state. *International Journal of Interactive Multimedia and Artificial Intelligence*, IP, p. 1, 12 2019.

OPPENHEIM, A. V.; SCHAFER, R. W. *Discrete-time signal processing*. [S.l.]: Prentice-Hall, 1999.

ORACLE. *Machine learning: conceito e aplicações*. 2025. Disponível em: ⟨https://www.oracle.com/br/artificial-intelligence/machine-learning/what-is-machine-learning/#:~:text=O%20machine%20learning%20(ML)%20%C3%A9,que%20imitam%20a%20intelig%C3%AAncia%20humana.⟩. Acesso em: 31 mar. 2025.

PAULINO, J. A. S. et al. Analysis of frequency range effect on the detection of voice disorder using convolutional neural networks trained on spectogram images. In: *2024 37th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. [S.l.: s.n.], 2024. p. 01–06.

PYCARET. *PyCaret documentation*. 2025. Disponível em: ⟨https://pycaret.readthedocs.io/⟩. Acesso em: 31 mar. 2025.

RABINER, L. R.; SCHAFER, R. W. *Theory and Application of Digital Signal Processing*. [S.l.]: Prentice-Hall, 2010.

RAWAT, P. et al. *A comprehensive study based on MFCC and spectrogram for audio classification*. 2023. Acesso em: 31 mar. 2025.

SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*, v. 6, n. 1, p. 60, 2019.

TITZE, I. R. *Principles of Voice Production*. Englewood Cliffs, NJ: Prentice Hall, 1994.

VIMAL, B. et al. Mfcc based audio classification using machine learning. In: *Proceedings of the 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. Kharagpur, Índia: IEEE, 2021.

WOLDERT-JOKISZ, B. *Saarbruecken Voice Database*. Saarbruecken: Institut für Phonetik, Universität des Saarlandes, 2007.