



SISTEMA DE RECOMENDAÇÃO DE CORES COM NAIVE BAYES

OBJETIVOS DO PROJETO

Teorema de Naive Bayes

Classificação e ajuste de paletas de cores

INTRODUÇÃO AO TEOREMA DE NAIVE BAYES

- O Naive Bayes é um classificador probabilístico baseado no teorema de Bayes.
- Ele usa a probabilidade condicional para classificar os dados, assumindo que as variáveis são independentes.
- Aprende, com os dados de treino, a probabilidade de cada atributo em cada classe.
- Usa essas probabilidades para calcular a chance de um novo exemplo pertencer a cada classe.
- Classifica na classe com maior probabilidade.

- Independência das Características: No seu caso, as cores podem ser tratadas como independentes, facilitando a análise.
- Simplicidade e Eficiência: O modelo é simples, rápido e adequado para problemas com múltiplas características (como cores) e muitas amostras.

INTRODUÇÃO AO TEOREMA DE NAIVE BAYES

FÓRMULA DE BAYES:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- $P(B | A)$ é a probabilidade de uma classe B dado um conjunto de características A.
- $P(A | B)$ é a probabilidade de observar A dado B.
- $P(B)$ é a probabilidade a priori da classe B.
- $P(A)$ é a probabilidade a priori das características A.

INTRODUÇÃO AO TEOREMA DE NAIVE BAYES

EXEMPLO DE USO

- Com a análise do algoritmo você consegue saber se uma paleta de cores é harmônica ou não.
- O Naive Bayes analisa os atributos (ex: média, desvio e distância entre cores).

Com base nas probabilidades aprendidas, ele estima:

- “Qual a chance dessa combinação ser harmônica?”
- “Qual a chance de não ser?”
- E escolhe a classe com maior chance.

IMPORT USO DE BIBLIOTECAS

sklearn (scikit-learn)

seaborn

Pillow (PIL)

joblib

numpy

itertools

pandas

re

matplotlib

tkinter

scikit-image





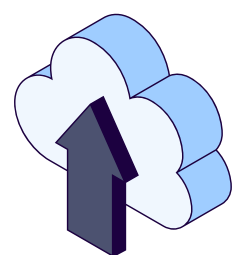
color-harmony-helper-rgb-para-lab/

└── Datasets/	Contém os datasets utilizados pelo sistema
└── dataset_rgb_rotulado1.csv	Arquivo CSV com valores RGB e rótulos (harmônico / não harmônico)
└── Modelo/	Contém o modelo treinado e outros artefatos do sistema
└── modelo_naive_bayes.pkl	Modelo Naive Bayes treinado e salvo (pode ser carregado para uso)
└── scaler.pkl	Arquivo com o scaler (normalizador) treinado
└── selector.pkl	Arquivo com o selector (selecionador de features) treinado
└── README.md	Descrição do projeto, como executá-lo e detalhes importantes
└── Scripts/	Contém os scripts Python principais
└── classificar_imagem.py	Script responsável por classificar as imagens com base nas cores
└── naive-bayes.py	Script principal para implementar o Naive Bayes e treinar o modelo
└── rgb-dataset-generator.py	Script utilizado para gerar datasets RGB

ESTRUTURA DA PASTA DO PROJETO

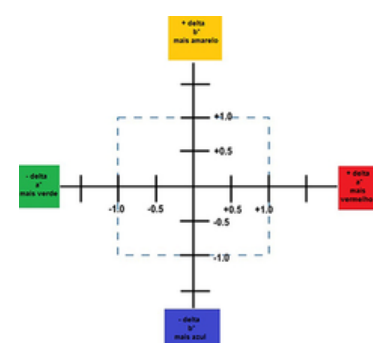
ETAPAS DO SISTEMA

1º



Leitura e
Agrupamento dos
Dados

2º



Conversão de RGB
para LAB

3º



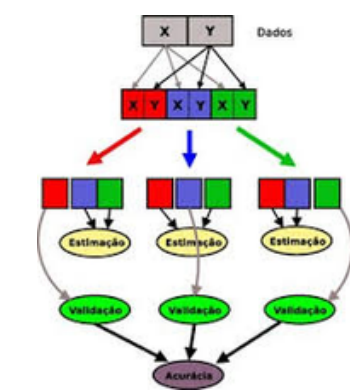
Extração das
Características

4º

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Aplicação do Naive
Bayes

5º



Validação Cruzada

6º



Curva de
Aprendizado

7º

		Valor Predito	
		Sim	Não
Real	Sim	Verdadeiro Positivo (TP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (TN)

Matriz de Confusão

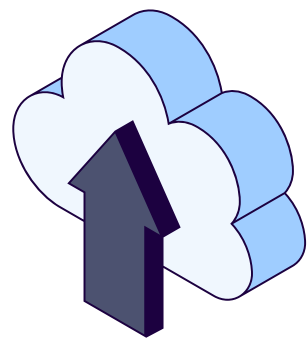
8º



Avaliação Final do
Modelo

LEITURA E AGRUPAMENTO DOS DADOS

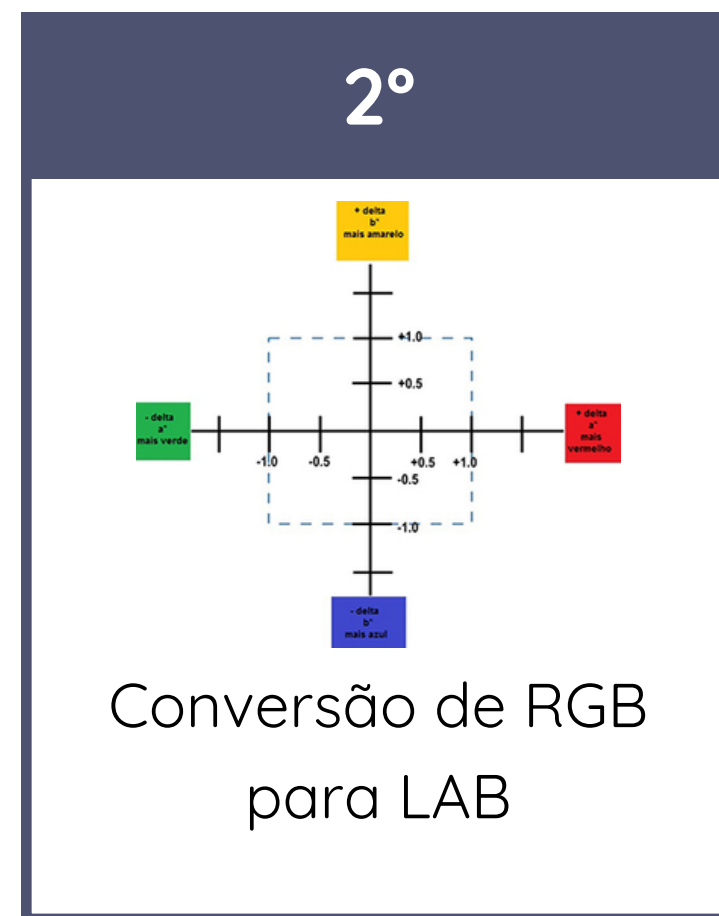
1º



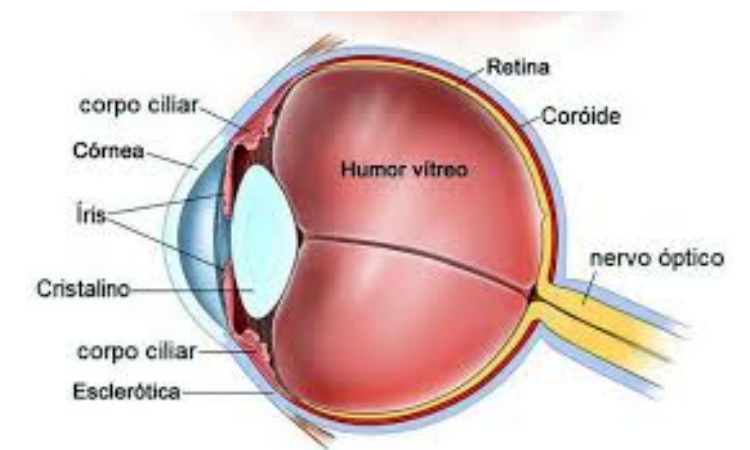
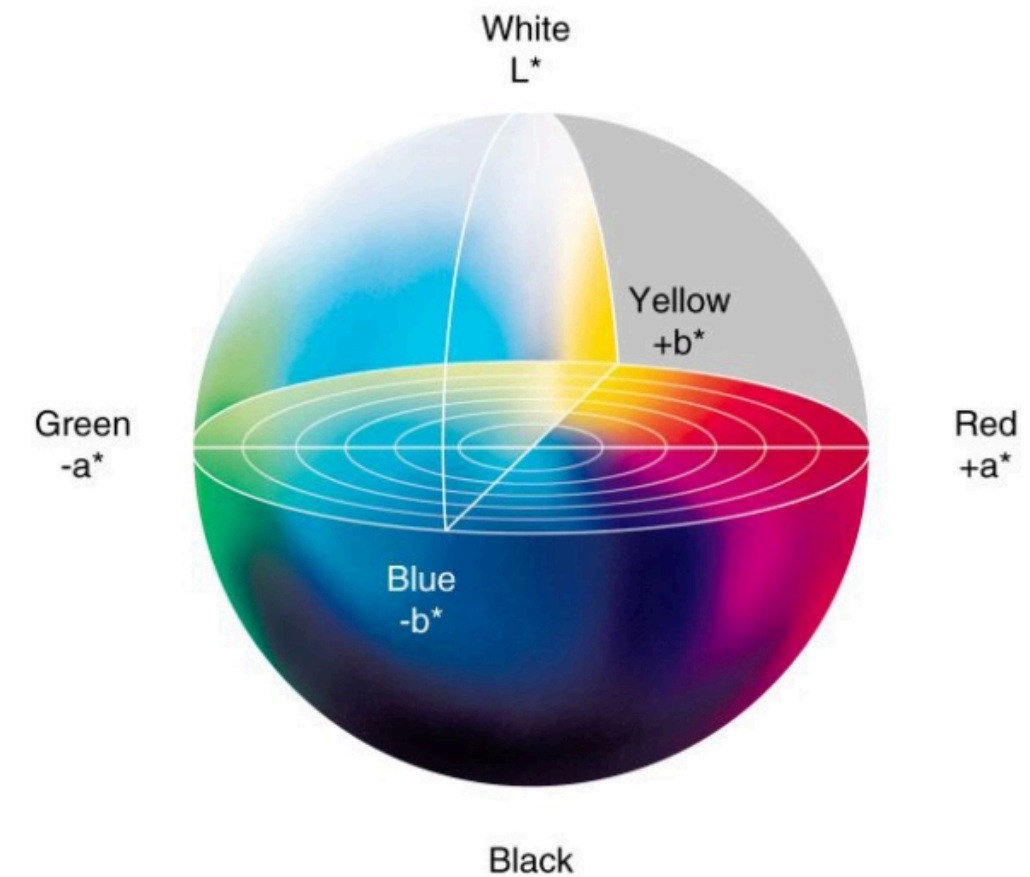
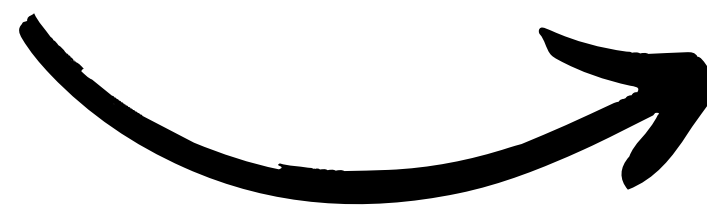
Leitura e
Agrupamento dos
Dados

HARMÔNICO X NÃO HARMÔNICO

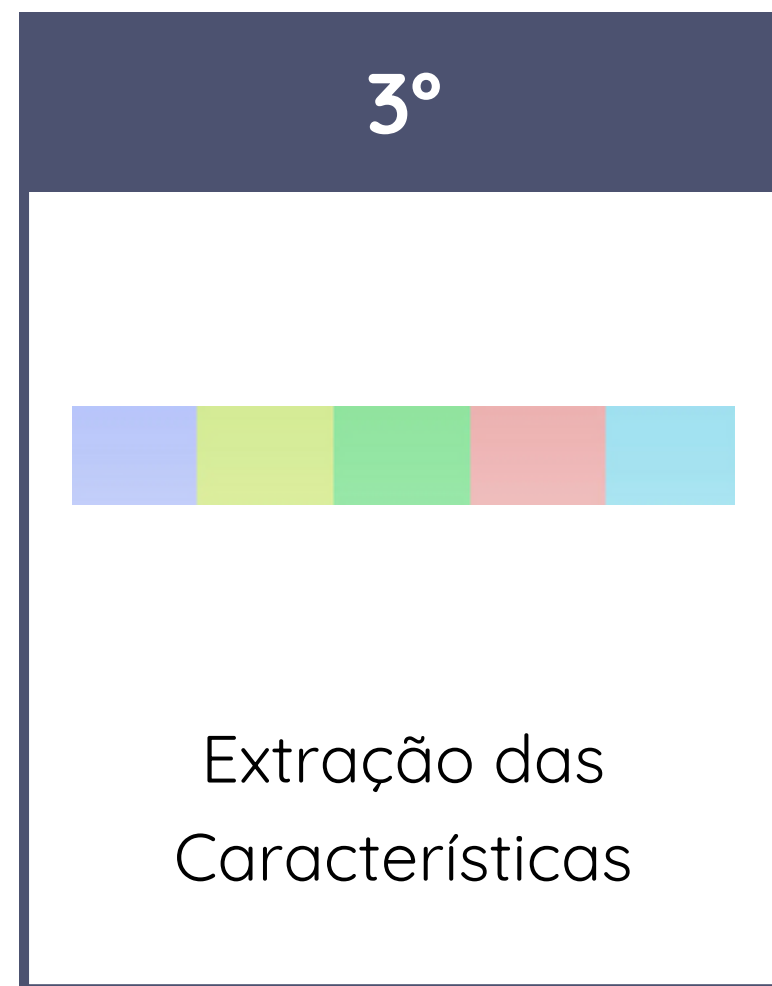
CONVERSÃO DE RGB PARA LAB



RGB

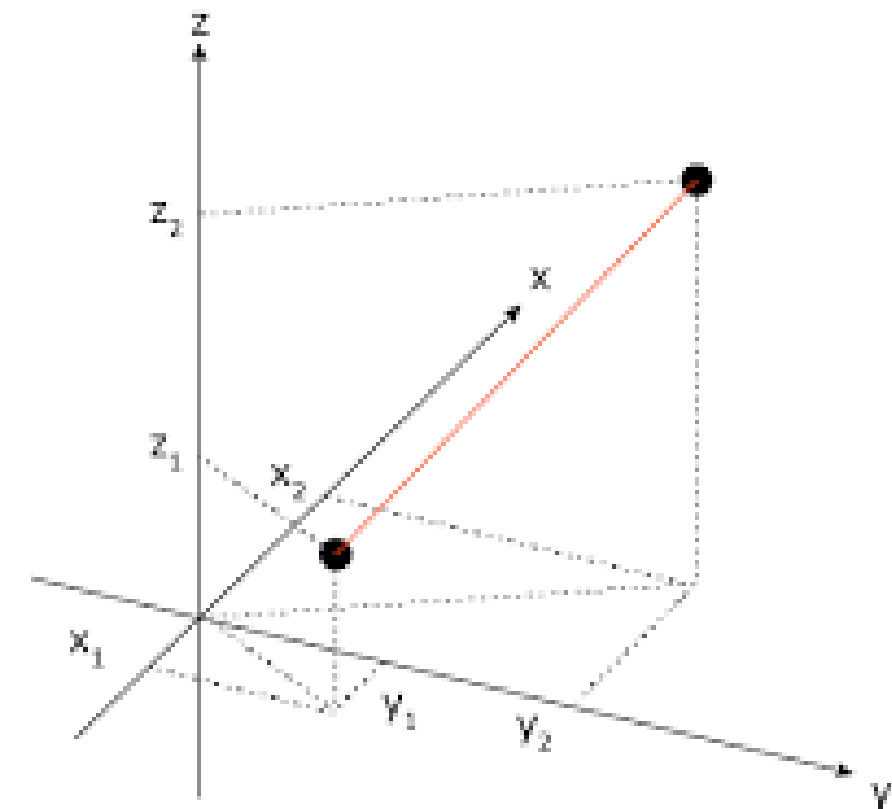


EXTRAÇÃO DAS CARACTERÍSTICAS



L, A, B

+



APLICAÇÃO DO NAIVE BAYES

4º

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

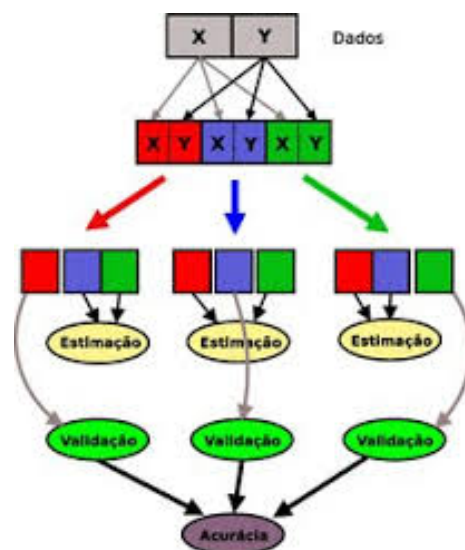
Aplicação do Naive
Bayes

O modelo Naive Bayes é treinado com as características extraídas (médias, desvios padrões, distâncias) e os rótulos ("harmônico" ou "não harmônico").

A validação cruzada e a validação de desempenho com a curva de aprendizado ajudam a avaliar a eficácia do modelo.

VALIDAÇÃO CRUZADA

5°



Validação Cruzada

Validação Cruzada:

É uma técnica para avaliar a performance do modelo de forma mais confiável, dividindo o conjunto de dados em várias partes (ou "folds").

1. Divide-se o dataset em 5 partes (folds).
 2. O modelo é treinado 5 vezes:
 - Em cada rodada, 1 parte é usada como teste, e as outras 4 como treino.
 3. Mede-se a acurácia/precisão em cada rodada.
 4. No fim, calcula-se a média e o desvio padrão dos resultados.
- **Evita resultados enganosos** que podem ocorrer ao treinar e testar no mesmo conjunto.
 - Garante que **todo o conjunto de dados foi usado** para treino e teste, mas em momentos diferentes.

CURVA DE APRENDIZADO

6º



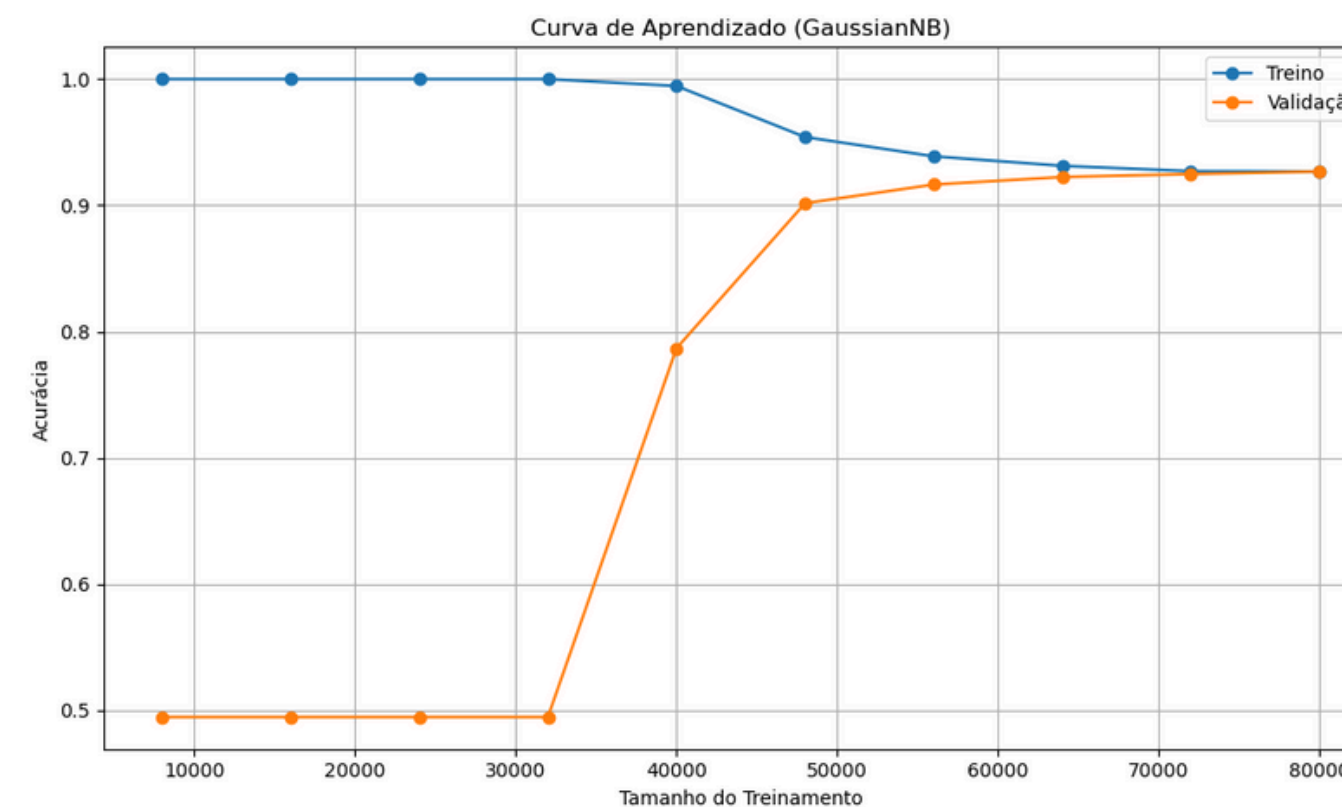
Curva de
Aprendizado

É um gráfico que mostra como a performance do modelo muda com a quantidade de dados de treino.

Mede-se o desempenho em:

Dados de treinamento (quão bem ele aprende)

Dados de validação (quão bem ele generaliza)



MATRIZ DE CONFUSÃO

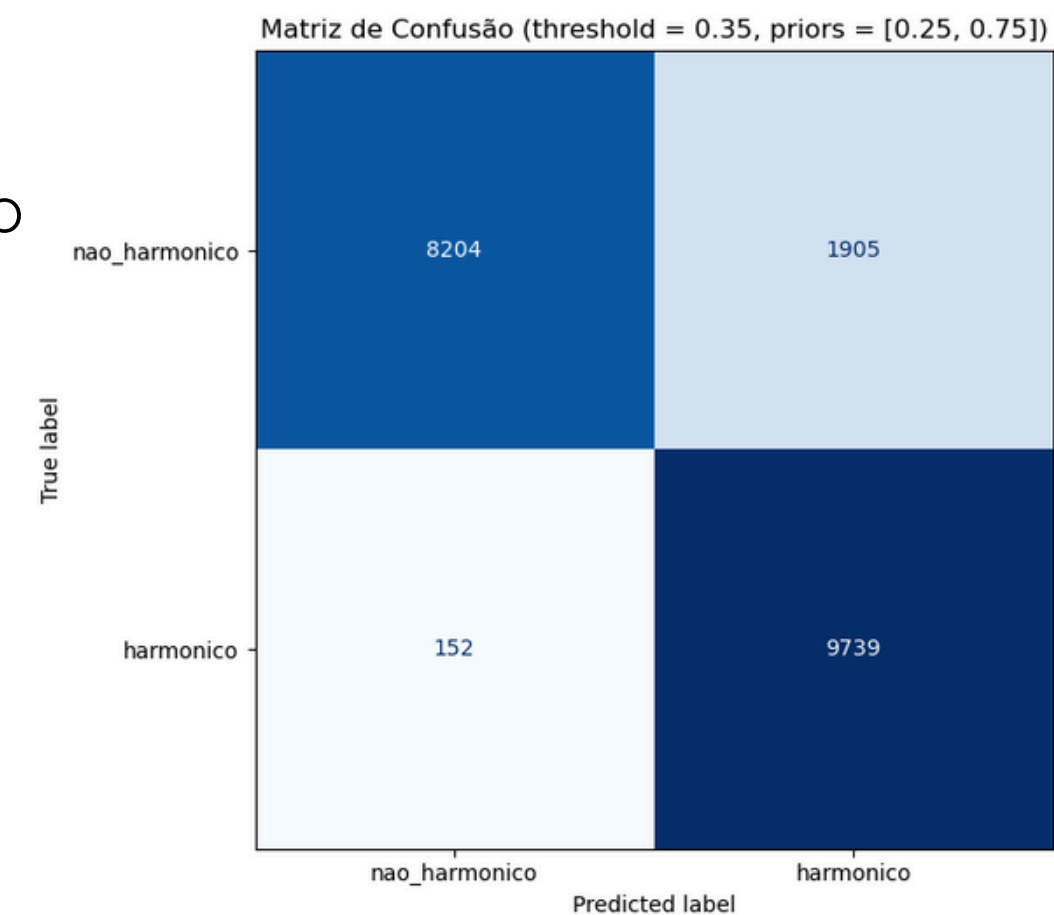
7º

		Valor Predito	
		Sim	Não
Real	Sim	Verdadeiro Positivo (TP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (TN)

Matriz de Confusão

É uma tabela que resume as previsões do modelo comparadas com os valores reais. Ela mostra onde o modelo acertou e onde errou. Usada para visualizar a precisão do modelo nas classificações.

A acurácia do modelo é medida usando a validação cruzada e a curva de aprendizado.



AVALIAÇÃO FINAL DO MODELO

CONCLUSÃO

RESULTADOS E MELHORIAS