

UNIVERSIDADE FEDERAL DO PARANÁ

LUCAS DE SOUZA RODRIGUES

APRENDIZADO DE MÁQUINA APLICADO À CLASSIFICAÇÃO DE EVENTOS DE  
APNEIA DO SONO

CURITIBA

2024

LUCAS DE SOUZA RODRIGUES

APRENDIZADO DE MÁQUINA APLICADO À CLASSIFICAÇÃO DE EVENTOS DE  
APNEIA DO SONO

Dissertação de mestrado apresentada ao programa de Pós- Graduação em Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Leandro dos Santos Coelho

CURITIBA

2024

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)  
UNIVERSIDADE FEDERAL DO PARANÁ  
SISTEMA DE BIBLIOTECAS – BIBLIOTECA DE CIÊNCIA E TECNOLOGIA

Rodrigues, Lucas de Souza

Aprendizado de máquina aplicado à classificação de eventos de apneia do sono / Lucas de Souza Rodrigues. – Curitiba, 2024.

1 recurso on-line : PDF.

Dissertação (Mestrado) - Universidade Federal do Paraná, Setor de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica.

Orientador: Leandro dos Santos Coelho

1. Apneia Obstrutiva do Sono. 2. Eletrocardiografia. 3. Redes Neurais de Computação. I. Universidade Federal do Paraná. II. Programa de Pós-Graduação em Engenharia Elétrica. III. Coelho, Leandro dos Santos. IV . Título.

Bibliotecário: Leticia Priscila Azevedo de Sousa CRB-9/2029



## TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação ENGENHARIA ELÉTRICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **LUCAS DE SOUZA RODRIGUES** intitulada: **APRENDIZADO DE MÁQUINA APLICADO À CLASSIFICAÇÃO DE EVENTOS DE APNEIA DO SONO**, sob orientação do Prof. Dr. LEANDRO DOS SANTOS COELHO, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 17 de Junho de 2024.

Assinatura Eletrônica

02/07/2024 11:24:46.0

LEANDRO DOS SANTOS COELHO  
Presidente da Banca Examinadora

Assinatura Eletrônica

09/07/2024 18:14:15.0

ROBERTO ZANETTI FREIRE  
Avaliador Externo (PONTIFICA UNIVERSIDADE CATÓLICA DO  
PARANÁ)

Assinatura Eletrônica

03/07/2024 17:44:12.0

LUIS HENRIQUE ASSUMPTÃO LOLIS  
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica

02/07/2024 11:01:11.0

GIDEON VILLAR LEANDRO  
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

*Dedico esta pesquisa ao meu pai, minha mãe e minha irmã.*

## **AGRADECIMENTOS**

Em primeiro lugar, eu quero muito de agradecer ao meu pai Roberto, minha mãe Eneida e minha irmã Carol. Sem o apoio, incentivo e ajuda deles eu jamais teria sequer tentado. Obrigado por toda paciência, conselhos, apoio e tudo mais que não dá para descrever que vocês me deram. Em segundo lugar, gostaria de agradecer à minha namorada Camille. Camille, você foi uma força motivadora e impulsionadora espetacular, fazendo muita coisa sair do lugar e andar. Você também foi paciente e carinhosa nos momentos mais difíceis, me ajudando a focar e manter a cabeça no lugar. Em terceiro lugar eu agradeço de forma muito especial ao meu orientador professor Leandro dos Santos Coelho, que acatou e defendeu a ideia do projeto e me ajudou com dicas, sugestões, críticas e muito conhecimento, nunca deixando o fator remoto ser um empecilho. Muito obrigado pela orientação!

Eu ainda gostaria de agradecer o apoio que nunca faltou dos meus amigos, das minhas tias e toda a minha família. São tantos que seria injusto citar alguns enquanto outros não, mas eu posso dizer que meus amigos do núcleo de Carajás, Belo Horizonte, São Paulo, Curitiba e mais, vocês foram fundamentais no incentivo e ajudas. Queria também agradecer ao apoio dos meus chefes que eu tive durante a realização do trabalho, sem o suporte dele jamais haveria conseguido conciliar a vida profissional com este trabalho. Em especial, meu antigo chefe Thiago Scaldaferrri que foi o que mais apoiou a ideia. Não queria deixar de citar e agradecer meus colegas de trabalho e as pessoas que me mostraram e incentivaram o caminho acadêmico, meus orientadores da época da graduação Franklin, Cristina e Murilo. Por fim, mas não menos importante, eu gostaria de agradecer a ajuda fundamental prestada pela minha antiga professora Carmela Braga, que de última hora escreveu uma carta de recomendação para meu ingresso no PPGEE-UFPR.

*"A expectativa é o maior impedimento para viver:  
leva-nos para o amanhã e  
faz com que se perca o presente."  
(SÊNECA, Sobre a brevidade da vida)*

*"A beleza não está nem na luz da manhã nem na sombra da noite,  
está no crepúsculo, nesse meio tom, nessa incerteza."  
(LYGIA FAGUNDES TELLES, Os contos)*

*"Devemos considerar mais de uma hipótese. Se alguma coisa deve  
ser explicada é preciso pensar em todas as maneiras  
diferentes pelas quais poderia ser explicada. Depois devemos pensar  
nos testes que poderiam servir para invalidar  
sistematicamente cada uma das alternativas."  
(CARL SAGAN, O mundo assombrado pelos demônios, cap. 12)*

## RESUMO

De acordo com dados estimados em 2019, a apneia obstrutiva do sono (AOS) é um distúrbio que afinge quase 1 bilhão de pessoas em todo o mundo. Uma parte dessas pessoas segue sem um diagnóstico e, conseqüentemente, sem um tratamento adequado. Isso ocorre pois o diagnóstico é feito após o paciente realizar um exame de polissonografia, que é de certa forma é um exame caro e inconveniente. Deste modo, tem-se buscado formas de auxiliar o diagnóstico da apneia por meio de ferramentas que dependam menos de especialistas e de sinais fisiológicos não invasivos. No contexto, a utilização de ferramentas de modelagem por aprendizado de máquina (ML) e aprendizado profundo (DL) com uso do eletrocardiograma (ECG) tem-se mostrado promissor. Diferentes estudos aplicaram diferentes tipos de modelo para classificar episódios de apneia do sono no ECG, variando os métodos de processar o sinal, os modelos empregados e a forma de avaliação. Esta dissertação de mestrado visa, então, a criação de um *framework* que auxilie no fluxo de desenvolvimento e teste de modelos de ML e DL na classificação de eventos de AOS. O fluxo envolve o processamento do sinal de ECG de canal único, a extração de características, a redução de dimensionalidade, o treinamento, validação, otimização, teste, avaliação de desempenho e explicabilidade dos classificadores treinados. Diferentemente de outros trabalhos, este *framework* visa aprofundar na seleção e importância dos hiperparâmetros e explicabilidade dos modelos utilizados. Este *framework* será aplicado na base de dados pública da PhysioNet. Ao todo foram avaliados 11 modelos de ML e 3 de DL. Com melhor dos modelos ML com máquinas de vetores de suporte (SVM, do inglês *support vector machines*) no experimento com análise de componentes principais (PCA, do inglês *principal component analysis*), com acurácia de 77,2%, precisão de 80,0%, sensibilidade de 84,3% e F1-Score igual a 0,821. O melhor modelo de aprendizado profundo foi a rede neural convolucional (CNN, do inglês *convolutional neural network*) com acurácia de 74,0%, sensibilidade de 77,9% e F1-Score de 0,788. O estudo de explicabilidade desta dissertação mostrou que as características do domínio da frequência impactaram mais a decisão do modelo. Os experimentos com redução de dimensionalidade desta dissertação também mostraram um efeito de melhora para distinguir a classe de eventos de não-apneia, enquanto que o uso de todas as características ajudou na detecção da classe de eventos de apneia.

**Palavras-chaves:** Apneia Obstrutiva do Sono; Aprendizado de Máquina; Aprendizado Profundo; Classificação; Eletrocardiograma; Redução de Dimensionalidade; Otimização de Hiperparâmetros; Explicabilidade.

## ABSTRACT

According to estimated data from 2019, obstructive sleep apnea (OSA) is a disorder that affects nearly 1 billion people worldwide. A portion of these individuals remains undiagnosed and, consequently, untreated. This is because the diagnosis is made after the patient undergoes a polysomnography examination, which is somewhat expensive and inconvenient. Thus, efforts have been made to assist in the diagnosis of sleep apnea through tools that rely less on specialists and non-invasive physiological signals. In this context, the use of machine learning (ML) and deep learning (DL) modeling tools with the use of electrocardiography (ECG) has shown promise. Different studies have applied different types of models to classify episodes of sleep apnea in ECG, varying the methods of signal processing, the models employed, and the evaluation approach. This master's thesis aims to create a framework that assists in the development and testing flow of ML and DL models in classifying OSA events. The flow involves processing single-channel ECG signals, feature extraction, dimensionality reduction, training, validation, optimization, testing, performance evaluation, and explainability of trained classifiers. Unlike other works, this framework aims to delve into the selection and importance of hyperparameters and explainability of the models used. This computational environment will be applied to the PhysioNet public database. Altogether, 11 ML models and 3 DL models were evaluated. The best ML model was support vector machine (SVM) in the experiment with principal component analysis (PCA), with an accuracy of 77.2%, precision of 80.0%, sensitivity of 84.3%, and F1-Score of 0.821. The best deep learning model was the convolutional neural network (CNN) with an accuracy of 74.0%, sensitivity of 77.9%, and F1-Score of 0.788. The explainability study present in this master's thesis conducted in this master's thesis has showed that frequency domain characteristics had the most impact on the model's decision. Dimensionality reduction experiments showed an improvement effect in distinguishing the non-apnea event class, while using all features aided in detecting the apnea event class.

**Key-words:** Obstructive Sleep Apnea. Machine Learning. Deep Learning. Classification. Electrocardiogram. Dimensionality Reduction. Hyperparameter Optimization. Explainability.

## LISTA DE ILUSTRAÇÕES

FIGURA 1 – METODOLOGIA DE PESQUISA . . . . .	32
FIGURA 2 – ESTÁGIOS DO SONO . . . . .	35
FIGURA 3 – EXEMPLO DE APARELHO CPAP . . . . .	42
FIGURA 4 – EXEMPLO DE APARELHO ORAL DE AVANÇO MANDIBULAR . . . . .	42
FIGURA 5 – COMPARATIVO ENTRE PROGRAMAÇÃO TRADICIONAL E APRENDIZADO DE MÁQUINA . . . . .	44
FIGURA 6 – TIPOS DE APRENDIZADO COM EXEMPLOS DE TÉCNICAS E PROBLEMAS APLICÁVEIS . . . . .	48
FIGURA 7 – FLUXO DE APRENDIZADO . . . . .	50
FIGURA 8 – MATRIZ DE CONFUSÃO . . . . .	51
FIGURA 9 – COMPARATIVO ENTRE AS CRUVAS DE REGRESSÃO LINEAR E LOGÍSTICA . . . . .	55
FIGURA 10 – ESTRUTURA DE UMA MÁQUINA DE VETOR DE SUPORTE . . . . .	56
FIGURA 11 – TIPOS DE PROBLEMAS BINÁRIOS . . . . .	57
FIGURA 12 – DISTÂNCIA DA MARGEM DE SEPARAÇÃO DO HIPERPLANO DA SVM . . . . .	58
FIGURA 13 – EXEMPLO DE APLICAÇÃO DO TRUQUE DO KERNEL . . . . .	59
FIGURA 14 – MODELO DO TIPO MULTICAMADAS DE <i>PERCEPTRONS</i> . . . . .	62
FIGURA 15 – ESTRUTURA DE UMA ÁRVORE DE DECISÃO . . . . .	64
FIGURA 16 – ESTRUTURA DE UMA FLORESTA ALEATÓRIA . . . . .	66
FIGURA 17 – ÁRVORE DE DECISÃO TIPO TRONCO . . . . .	68
FIGURA 18 – <i>ADAPTIVE BOOSTING</i> . . . . .	69
FIGURA 19 – ESTRUTURA DO GRADIENT BOOSTING . . . . .	70
FIGURA 20 – ESTRUTURA DE APRENDIZADO POR CONJUNTO . . . . .	71
FIGURA 21 – EXEMPLO DE VOTO MAJORITÁRIO COM SISTEMA MAJORITÁRIO . . . . .	73
FIGURA 22 – EXEMPLO DE UMA TRANSFORMAÇÃO APLICADA PELO MÉTODO T-SNE . . . . .	77
FIGURA 23 – EXEMPLO DA PRIMEIRA ITERAÇÃO DO ALGORITMO T-SNE COM CÁLCULO DA MATRIZ DE SIMILARIDADE . . . . .	78
FIGURA 24 – EXEMPLO DE CÁLCULO DA SIMILARIDADE NO MÉTODO UMAP . . . . .	80
FIGURA 25 – EXEMPLO DE AJUSTE DOS DADOS NO ESPAÇO TRANSFORMADO NO MÉTODO UMAP . . . . .	81
FIGURA 26 – VISÃO GERAL DO SISTEMA OPTUNA . . . . .	84
FIGURA 27 – CONTRIBUIÇÃO MARGINAL DO VALOR SHAPLEY . . . . .	86

FIGURA 28 – CONTRIBUIÇÃO MARGINAL DO VALOR SHAPLEY DAS COMBINAÇÕES COM E SEM JOGADOR 1 . . . . .	86
FIGURA 29 – MÉDIA DAS DIFERENÇAS DAS CONTRIBUIÇÕES DAS COMBINAÇÕES COM E SEM JOGADOR 1 . . . . .	87
FIGURA 30 – EXEMPLO DE OPERAÇÃO NA CAMADA DE CONVOLUÇÃO . . . . .	91
FIGURA 31 – EXEMPLO DE UMA CNN . . . . .	92
FIGURA 32 – EXEMPLO DE UMA RNN . . . . .	93
FIGURA 33 – ARQUITETURA DE UMA UNIDADE DE LSTM . . . . .	94
FIGURA 34 – BASE SCOPUS . . . . .	97
FIGURA 35 – BASE WOS . . . . .	98
FIGURA 36 – VOS SCOPUS . . . . .	104
FIGURA 37 – VOS WOS . . . . .	105
FIGURA 38 – EXEMPLO DE ECG COM MARCAÇÕES DA BASE DE DADOS . . . . .	114
FIGURA 39 – ECG DE PACIENTE DO GRUPO C NO MOMENTO DE RESPIRAÇÕES NORMAIS . . . . .	120
FIGURA 40 – ECG DE PACIENTE DO GRUPO A NO MOMENTO DE APNEIA . . . . .	121
FIGURA 41 – MACRO FLUXO DE PROCESSAMENTO PARA MODELOS DE ML	121
FIGURA 42 – MACRO FLUXO DE PROCESSAMENTO PARA MODELOS DE DL	121
FIGURA 43 – SÉRIE R-R DE PACIENTE DO GRUPO C NO MOMENTO DE RESPIRAÇÕES NORMAIS . . . . .	123
FIGURA 44 – ECG DE PACIENTE DO GRUPO A NO MOMENTO DE APNEIA . . . . .	123
FIGURA 45 – ARQUITETURA PROPOSTA PARA CNN . . . . .	137
FIGURA 46 – IMPORTÂNCIA GERAL DE CADA DOMÍNIO DE CARACTERÍSTICAS PARA O EXPERIMENTO I . . . . .	147
FIGURA 47 – IMPORTÂNCIA GERAL DE CADA DE CARACTERÍSTICAS PARA O EXPERIMENTO I . . . . .	148
FIGURA 48 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO ADABOOST NO EXPERIMENTO I . . . . .	149
FIGURA 49 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DE UM EXEMPLO DO MODELO ADABOOST . . . . .	151
FIGURA 50 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DE OITO CASOS DO MODELO ADABOOST . . . . .	152
FIGURA 51 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO ADABOOST . . . . .	153
FIGURA 52 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO SVM NO EXPERIMENTO I . . . . .	155

FIGURA 53 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO SVM NO EXPERIMENTO II . . . . .	156
FIGURA 54 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO SVM NO EXPERIMENTO III . . . . .	157
FIGURA 55 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO SVM NO EXPERIMENTO IV . . . . .	157
FIGURA 56 – IMPACTOS DOS HIPERPARÂMETROS NA OTIMIZAÇÃO DO MODELO SVM . . . . .	171
FIGURA 57 – MATRIZ DE CONFUSÃO DO MODELO SVM . . . . .	172
FIGURA 58 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO CNN . . . . .	172
FIGURA 59 – IMPACTO RELATIVO DE CADA HIPERPARÂMETRO NO DESEMPENHO E NO TEMPO DO MODELO DO CNN . . . . .	173
FIGURA 60 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO ADABOOST NO EXPERIMENTO I . . . . .	208
FIGURA 61 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO ADABOOST NO EXPERIMENTO II . . . . .	208
FIGURA 62 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO ADABOOST NO EXPERIMENTO III . . . . .	209
FIGURA 63 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO ADABOOST NO EXPERIMENTO IV . . . . .	209
FIGURA 64 – IMPACTOS DOS HIPERPARÂMETROS NA OTIMIZAÇÃO DO MODELO ADABOOST . . . . .	210
FIGURA 65 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO ADABOOST NO EXPERIMENTO II . . . . .	211
FIGURA 66 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO ADABOOST NO EXPERIMENTO III . . . . .	212
FIGURA 67 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO ADABOOST NO EXPERIMENTO IV . . . . .	213
FIGURA 68 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO ADABOOST NO EXPERIMENTO II . . . . .	214
FIGURA 69 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO ADABOOST NO EXPERIMENTO III . . . . .	214
FIGURA 70 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO ADABOOST NO EXPERIMENTO IV . . . . .	214
FIGURA 71 – MATRIZ DE CONFUSÃO DO MODELO ADABOOST . . . . .	215

FIGURA 72 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO MLP NO EXPERIMENTO I . . . . .	215
FIGURA 73 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO MLP NO EXPERIMENTO II . . . . .	216
FIGURA 74 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO MLP NO EXPERIMENTO III . . . . .	216
FIGURA 75 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO MLP NO EXPERIMENTO IV . . . . .	217
FIGURA 76 – IMPACTOS DOS HIPERPARÂMETROS NA OTIMIZAÇÃO DO MODELO MLP . . . . .	218
FIGURA 77 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO MLP NO EXPERIMENTO I . . . . .	219
FIGURA 78 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO MLP NO EXPERIMENTO II . . . . .	220
FIGURA 79 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO MLP NO EXPERIMENTO III . . . . .	221
FIGURA 80 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO MLP NO EXPERIMENTO IV . . . . .	222
FIGURA 81 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO MLP NO EXPERIMENTO I . . . . .	223
FIGURA 82 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO MLP NO EXPERIMENTO II . . . . .	223
FIGURA 83 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO MLP NO EXPERIMENTO III . . . . .	224
FIGURA 84 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO MLP NO EXPERIMENTO IV . . . . .	224
FIGURA 85 – MATRIZ DE CONFUSÃO DO MODELO MLP . . . . .	224
FIGURA 86 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO <i>BAGGING</i> NO EXPERIMENTO I . . . . .	225
FIGURA 87 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO <i>BAGGING</i> NO EXPERIMENTO II . . . . .	225
FIGURA 88 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO <i>BAGGING</i> NO EXPERIMENTO III . . . . .	226
FIGURA 89 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO <i>BAGGING</i> NO EXPERIMENTO IV . . . . .	226

FIGURA 90 – IMPACTOS DOS HIPERPARÂMETROS NA OTIMIZAÇÃO DO MODELO <i>BAGGING</i> . . . . .	227
FIGURA 91 – MATRIZ DE CONFUSÃO DO MODELO <i>BAGGING</i> . . . . .	228
FIGURA 92 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO DT NO EXPERIMENTO I . . . . .	228
FIGURA 93 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO DT NO EXPERIMENTO II . . . . .	229
FIGURA 94 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO DT NO EXPERIMENTO III . . . . .	229
FIGURA 95 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO DT NO EXPERIMENTO IV . . . . .	230
FIGURA 96 – IMPACTOS DOS HIPERPARÂMETROS NA OTIMIZAÇÃO DO MODELO DT . . . . .	231
FIGURA 97 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO DT NO EXPERIMENTO I . . . . .	232
FIGURA 98 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO DT NO EXPERIMENTO II . . . . .	233
FIGURA 99 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO DT NO EXPERIMENTO III . . . . .	234
FIGURA 100–IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO DT NO EXPERIMENTO IV . . . . .	235
FIGURA 101–CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO DT NO EXPERIMENTO I . . . . .	236
FIGURA 102–CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO DT NO EXPERIMENTO II . . . . .	236
FIGURA 103–CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO DT NO EXPERIMENTO III . . . . .	237
FIGURA 104–CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO DT NO EXPERIMENTO IV . . . . .	237
FIGURA 105–MATRIZ DE CONFUSÃO DO MODELO DT . . . . .	237
FIGURA 106–HISTÓRICO DE OTIMIZAÇÃO DO MODELO VM NO EXPERIMENTO I . . . . .	238
FIGURA 107–HISTÓRICO DE OTIMIZAÇÃO DO MODELO VM NO EXPERIMENTO II . . . . .	239

FIGURA 108–HISTÓRICO DE OTIMIZAÇÃO DO MODELO VM NO EXPERIMENTO III . . . . .	239
FIGURA 109–HISTÓRICO DE OTIMIZAÇÃO DO MODELO VM NO EXPERIMENTO IV . . . . .	240
FIGURA 110–MATRIZ DE CONFUSÃO DO MODELO VM . . . . .	242
FIGURA 111–HISTÓRICO DE OTIMIZAÇÃO DO MODELO LIGHTGBM NO EXPERIMENTO I . . . . .	243
FIGURA 112–HISTÓRICO DE OTIMIZAÇÃO DO MODELO LIGHTGBM NO EXPERIMENTO II . . . . .	243
FIGURA 113–HISTÓRICO DE OTIMIZAÇÃO DO MODELO LIGHTGBM NO EXPERIMENTO III . . . . .	244
FIGURA 114–HISTÓRICO DE OTIMIZAÇÃO DO MODELO LIGHTGBM NO EXPERIMENTO IV . . . . .	244
FIGURA 115–MATRIZ DE CONFUSÃO DO MODELO LIGHTGBM . . . . .	246
FIGURA 116–HISTÓRICO DE OTIMIZAÇÃO DO MODELO LR NO EXPERIMENTO I . . . . .	247
FIGURA 117–HISTÓRICO DE OTIMIZAÇÃO DO MODELO LR NO EXPERIMENTO II . . . . .	247
FIGURA 118–HISTÓRICO DE OTIMIZAÇÃO DO MODELO LR NO EXPERIMENTO III . . . . .	248
FIGURA 119–HISTÓRICO DE OTIMIZAÇÃO DO MODELO LR NO EXPERIMENTO IV . . . . .	248
FIGURA 120–IMPACTOS DOS HIPERPARÂMETROS NA OTIMIZAÇÃO DO MODELO LR . . . . .	249
FIGURA 121–IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO LR NO EXPERIMENTO I . . . . .	250
FIGURA 122–IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO LR NO EXPERIMENTO II . . . . .	251
FIGURA 123–IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO LR NO EXPERIMENTO III . . . . .	252
FIGURA 124–IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO LR NO EXPERIMENTO IV . . . . .	253
FIGURA 125–CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO LR NO EXPERIMENTO I . . . . .	254
FIGURA 126–CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO LR NO EXPERIMENTO II . . . . .	254

FIGURA 127–CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO LR NO EXPERIMENTO III . . . . .	255
FIGURA 128–CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO LR NO EXPERIMENTO IV . . . . .	256
FIGURA 129–MATRIZ DE CONFUSÃO DO MODELO LR . . . . .	256
FIGURA 130–IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO NB NO EXPERIMENTO I . . . . .	257
FIGURA 131–IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO NB NO EXPERIMENTO II . . . . .	258
FIGURA 132–IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO NB NO EXPERIMENTO III . . . . .	259
FIGURA 133–IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO NB NO EXPERIMENTO IV . . . . .	260
FIGURA 134–CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO NB NO EXPERIMENTO I . . . . .	261
FIGURA 135–CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO NB NO EXPERIMENTO II . . . . .	261
FIGURA 136–CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO NB NO EXPERIMENTO III . . . . .	262
FIGURA 137–CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO NB NO EXPERIMENTO IV . . . . .	262
FIGURA 138–MATRIZ DE CONFUSÃO DO MODELO NB . . . . .	262
FIGURA 139–HISTÓRICO DE OTIMIZAÇÃO DO MODELO RF NO EXPERIMENTO I . . . . .	263
FIGURA 140–HISTÓRICO DE OTIMIZAÇÃO DO MODELO RF NO EXPERIMENTO II . . . . .	263
FIGURA 141–HISTÓRICO DE OTIMIZAÇÃO DO MODELO RF NO EXPERIMENTO III . . . . .	264
FIGURA 142–HISTÓRICO DE OTIMIZAÇÃO DO MODELO RF NO EXPERIMENTO IV . . . . .	264
FIGURA 143–IMPACTOS DOS HIPERPARÂMETROS NA OTIMIZAÇÃO DO MODELO RF . . . . .	265

FIGURA 144–IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO RF NO EXPERIMENTO I . . . . .	266
FIGURA 145–IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO RF NO EXPERIMENTO II . . . . .	267
FIGURA 146–IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO RF NO EXPERIMENTO III . . . . .	268
FIGURA 147–IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO RF NO EXPERIMENTO IV . . . . .	269
FIGURA 148–CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO RF NO EXPERIMENTO I . . . . .	270
FIGURA 149–CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO RF NO EXPERIMENTO II . . . . .	270
FIGURA 150–CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO RF NO EXPERIMENTO III . . . . .	271
FIGURA 151–CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO RF NO EXPERIMENTO IV . . . . .	271
FIGURA 152–MATRIZ DE CONFUSÃO DO MODELO RF . . . . .	271
FIGURA 153–HISTÓRICO DE OTIMIZAÇÃO DO MODELO XGBOOST NO EXPERIMENTO I . . . . .	272
FIGURA 154–HISTÓRICO DE OTIMIZAÇÃO DO MODELO XGBOOST NO EXPERIMENTO II . . . . .	272
FIGURA 155–HISTÓRICO DE OTIMIZAÇÃO DO MODELO XGBOOST NO EXPERIMENTO III . . . . .	273
FIGURA 156–HISTÓRICO DE OTIMIZAÇÃO DO MODELO XGBOOST NO EXPERIMENTO IV . . . . .	273
FIGURA 157–IMPACTOS DOS HIPERPARÂMETROS NA OTIMIZAÇÃO DO MODELO XGBOOST . . . . .	274
FIGURA 158–IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO XGBOOST NO EXPERIMENTO I . . . . .	275
FIGURA 159–IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO XGBOOST NO EXPERIMENTO II . . . . .	276
FIGURA 160–IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO XGBOOST NO EXPERIMENTO III . . . . .	277
FIGURA 161–IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO XGBOOST NO EXPERIMENTO IV . . . . .	278

FIGURA 162—CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO XGBOOST NO EXPERIMENTO I . . . . .	279
FIGURA 163—CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO XGBOOST NO EXPERIMENTO II . . . . .	279
FIGURA 164—CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO XGBOOST NO EXPERIMENTO III . . . . .	280
FIGURA 165—CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO XGBOOST NO EXPERIMENTO IV . . . . .	280
FIGURA 166—MATRIZ DE CONFUSÃO DO MODELO XGBOOST . . . . .	280
FIGURA 167—HISTÓRICO DE OTIMIZAÇÃO PARA MODELO LSTM . . . . .	282
FIGURA 168—IMPORTÂNCIA RELATIVA NO RESULTADO E IMPACTO RELATIVO NO TEMPO DOS HIPERPARÂMETROS DURANTE OTIMIZAÇÃO DO MODELO LSTM . . . . .	283
FIGURA 169—HISTÓRICO DE OTIMIZAÇÃO PARA MODELO HÍBRIDO . . . . .	284
FIGURA 170—IMPORTÂNCIA RELATIVA NO RESULTADO E IMPACTO RELATIVO NO TEMPO DOS HIPERPARÂMETROS DURANTE OTIMIZAÇÃO DO MODELO HÍBRIDO . . . . .	285

## LISTA DE TABELAS

TABELA 1 – EXEMPLOS DE SINTOMAS CAUSADOS POR DISTÚRBIOS DO SONO . . . . .	38
TABELA 2 – USO DO IAH NA CLASSIFICAÇÃO DA SEVERIDADE DA AOS .	40
TABELA 3 – EXEMPLO DE VOTO MAJORITÁRIO COM SISTEMA PONDERADO	72
TABELA 4 – EXEMPLO DE REDUÇÃO DE REDUÇÃO DE DIMENSIONALIDADE COM PCA . . . . .	76
TABELA 5 – RESUMO DOS ARTIGOS SELECIONADOS - PARTE I . . . . .	99
TABELA 6 – RESUMO DOS ARTIGOS SELECIONADOS - PARTE II . . . . .	100
TABELA 7 – RESUMO DOS ARTIGOS SELECIONADOS - PARTE III . . . . .	101
TABELA 8 – GRUPO A DE PACIENTES DA BASE DE DADOS . . . . .	116
TABELA 9 – GRUPO B DE PACIENTES DA BASE DE DADOS . . . . .	117
TABELA 10 – GRUPO C DE PACIENTES DA BASE DE DADOS . . . . .	117
TABELA 11 – GRUPO X DE PACIENTES DA BASE DE DADOS . . . . .	119
TABELA 12 – PARÂMETROS UTILIZADOS NO FILTRO . . . . .	125
TABELA 13 – HIPERPARÂMETROS DAS TÉCNICAS DE REDUÇÃO DE DIMENSIONALIDADE . . . . .	129
TABELA 14 – Separação das amostras em treino, validação e teste . . . . .	129
TABELA 15 – CONFIGURAÇÃO DO PROCESSO DE OTIMIZAÇÃO E SINTONIA DE HIPERPARÂMETROS . . . . .	130
TABELA 16 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DO ADABOOST . . . . .	131
TABELA 17 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DO MLP . .	131
TABELA 18 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DO <i>BAGGING</i>	132
TABELA 19 – VALORES DOS PRINCIPAIS HIPERPARÂMETROS DOS CLASSIFICADORES USADOS NO MODELO DE <i>BAGGING</i> . . . . .	132
TABELA 20 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DE DT . . .	133
TABELA 21 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DO VM . .	134
TABELA 22 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DO LIGHTGBM	134
TABELA 23 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DO MLP . .	135
TABELA 24 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DA RF . . .	135
TABELA 25 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DA RF . . .	136
TABELA 26 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DO XGBOOST	136
TABELA 27 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DAS CNN .	138
TABELA 28 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DO LSTM .	138

TABELA 29 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DA REDE HÍBRIDA . . . . .	139
TABELA 30 – EXPERIMENTOS COM OS MODELOS DE ML . . . . .	140
TABELA 31 – RESUMO DA SINTONIA DOS HIPERPARÂMETROS DAS TÉCNICAS DE REDUÇÃO DE DIMENSIONALIDADE . . . . .	142
TABELA 32 – HIPERPARÂMETROS SELECIONADOS PARA A TÉCNICA T-SNE	143
TABELA 33 – HIPERPARÂMETROS SELECIONADOS PARA A TÉCNICA UMAP	143
TABELA 34 – RESULTADOS DAS MÉTRICAS DE DESEMPENHO OBTIDAS PARA OS MODELOS DE CLASSIFICAÇÃO SEM REDUÇÃO DE DIMENSIONALIDADE . . . . .	144
TABELA 35 – RESULTADOS DAS MÉTRICAS DE DESEMPENHO OBTIDAS PARA OS MODELOS DE CLASSIFICAÇÃO COM APLICAÇÃO DO PCA . . . . .	145
TABELA 36 – RESULTADOS DAS MÉTRICAS DE DESEMPENHO OBTIDAS PARA OS MODELOS DE CLASSIFICAÇÃO COM APLICAÇÃO DO T-SNE . . . . .	145
TABELA 37 – RESULTADOS DAS MÉTRICAS DE DESEMPENHO OBTIDAS PARA OS MODELOS DE CLASSIFICAÇÃO COM APLICAÇÃO DO UMAP . . . . .	145
TABELA 38 – TEMPO TOTAL DE EXECUÇÃO DOS EXPERIMENTOS . . . . .	146
TABELA 39 – HIPERPARÂMETROS SELECIONADOS NO MODELO SVM E TEMPO DE OTIMIZAÇÃO . . . . .	154
TABELA 40 – HIPERPARÂMETROS SELECIONADOS NO MODELO SVM E TEMPO DE OTIMIZAÇÃO . . . . .	156
TABELA 41 – TEMPOS DE TESTE E RESULTADO DOS TESTES DE HIPÓTESES DO MODELO SVM . . . . .	158
TABELA 42 – HIPERPARÂMETROS SELECIONADOS NO MODELO SVM E TEMPO DE OTIMIZAÇÃO . . . . .	159
TABELA 43 – HIPERPARÂMETROS SELECIONADOS NO MODELO MLP E TEMPO DE OTIMIZAÇÃO . . . . .	160
TABELA 44 – HIPERPARÂMETROS SELECIONADOS NO MODELO <i>BAGGING</i> E TEMPO DE OTIMIZAÇÃO . . . . .	161
TABELA 45 – HIPERPARÂMETROS SELECIONADOS NO MODELO DT E TEMPO DE OTIMIZAÇÃO . . . . .	161
TABELA 46 – HIPERPARÂMETROS SELECIONADOS NO MODELO VM E TEMPO DE OTIMIZAÇÃO NO EXPERIMENTO I . . . . .	162
TABELA 47 – HIPERPARÂMETROS SELECIONADOS NO MODELO VM E TEMPO DE OTIMIZAÇÃO NO EXPERIMENTO II . . . . .	162

TABELA 48 – HIPERPARÂMETROS SELECIONADOS NO MODELO VM E TEMPO DE OTIMIZAÇÃO NO EXPERIMENTO III . . . . .	163
TABELA 49 – HIPERPARÂMETROS SELECIONADOS NO MODELO VM E TEMPO DE OTIMIZAÇÃO NO EXPERIMENTO IV . . . . .	164
TABELA 50 – HIPERPARÂMETROS SELECIONADOS NO MODELO LIGHTGBM E TEMPO DE OTIMIZAÇÃO NO EXPERIMENTO I E II . . . . .	164
TABELA 51 – HIPERPARÂMETROS SELECIONADOS NO MODELO LIGHTGBM E TEMPO DE OTIMIZAÇÃO NO EXPERIMENTO III E IV . . . . .	165
TABELA 52 – HIPERPARÂMETROS SELECIONADOS NO MODELO LR E TEMPO DE OTIMIZAÇÃO . . . . .	165
TABELA 53 – HIPERPARÂMETROS SELECIONADOS NO MODELO RF E TEMPO DE OTIMIZAÇÃO . . . . .	166
TABELA 54 – HIPERPARÂMETROS SELECIONADOS NO MODELO XGBOOST E TEMPO DE OTIMIZAÇÃO . . . . .	167
TABELA 55 – RESULTADOS DE DESEMPENHO DOS MODELOS DE DL . . .	167
TABELA 56 – OUTROS RESULTADOS DOS MODELOS DE DL . . . . .	168
TABELA 57 – HIPERPARÂMETROS SELECIONADOS PARA O MODELO CNN	169
TABELA 58 – HIPERPARÂMETROS SELECIONADOS PARA O MODELO LSTM	169
TABELA 59 – HIPERPARÂMETROS SELECIONADOS PARA O MODELO HÍBRIDO DE CNN E LSTM . . . . .	170
TABELA 60 – COMPARATIVO ENTRE OS EXPERIMENTOS COM AS MÉDIAS DOS RESULTADOS OBTIDOS . . . . .	175
TABELA 61 – COMPARATIVO DE RESULTADOS COM TRABALHOS SIMILARES DA LITERATURA . . . . .	177
TABELA 62 – IMPACTO RELATIVO NO RESULTADO E NO TEMPO DE CADA HIPERPARÂMETRO DO MODELO VM NO EXPERIMENTO I . .	240
TABELA 63 – IMPACTO RELATIVO NO RESULTADO E NO TEMPO DE CADA HIPERPARÂMETRO DO MODELO VM NO EXPERIMENTO II . .	241
TABELA 64 – IMPACTO RELATIVO NO RESULTADO E NO TEMPO DE CADA HIPERPARÂMETRO DO MODELO VM NO EXPERIMENTO III . .	241
TABELA 65 – IMPACTO RELATIVO NO RESULTADO E NO TEMPO DE CADA HIPERPARÂMETRO DO MODELO VM NO EXPERIMENTO IV . .	242
TABELA 66 – IMPACTO RELATIVO NO RESULTADO E NO TEMPO DE CADA HIPERPARÂMETRO DO MODELO LIGHTGBM NO EXPERIMENTO I . . . . .	245
TABELA 67 – IMPACTO RELATIVO NO RESULTADO E NO TEMPO DE CADA HIPERPARÂMETRO DO MODELO LIGHTGBM NO EXPERIMENTO II . . . . .	245

TABELA 68 – IMPACTO RELATIVO NO RESULTADO E NO TEMPO DE CADA HIPERPARÂMETRO DO MODELO LIGHTGBM NO EXPERIMENTO III . . . . .	245
TABELA 69 – IMPACTO RELATIVO NO RESULTADO E NO TEMPO DE CADA HIPERPARÂMETRO DO MODELO LIGHTGBM NO EXPERIMENTO IV . . . . .	246
TABELA 70 – RESULTADOS DOS TESTES DE HIPÓTESES E TEMPOS DE PREDIÇÃO DA BASE DE TESTE PARA MODELOS ML . . . . .	281
TABELA 71 – RESULTADOS DOS TESTES DE HIPÓTESES E TEMPOS DE PREDIÇÃO DA BASE DE TESTE PARA MODELOS ML . . . . .	281

## LISTA DE ABREVIATURAS E DE SIGLAS

<b>ACS</b>	Apneia do Sono Central
<b>ACS</b>	Apneia Central do Sono
<b>AMD</b>	<i>Advanced Micro Devices</i>
<b>ANN</b>	<i>Artificial Neuron Networks</i>
<b>AOS</b>	Apneia do Sono Obstrutiva
<b>AdaBoost</b>	<i>Adaptive Boosting</i>
<b>AutoML</b>	<i>Automate Machine Learning</i>
<b>BoF</b>	<i>Bag-of-Features</i>
<b>CMA-ES</b>	<i>Covariance Matrix Adaptation-Evolution Strategy</i>
<b>CNN</b>	<i>Convolutional Neuron Networks</i>
<b>DL</b>	<i>Deep Learning</i>
<b>DT</b>	<i>Decision Trees</i>
<b>DVS</b>	Decomposição por Valores Singulares
<b>ECG</b>	Eletrocardiograma
<b>EDR</b>	Respiração Derivada do ECG
<b>EEG</b>	Eletroencefalograma
<b>FIR</b>	filtro <i>Finite Impulse Response</i>
<b>FN</b>	Falso Negativo, do inglês <i>False Negative</i>
<b>FP</b>	Falso Positivos, do inglês <i>False Positive</i>
<b>GAN</b>	<i>Generative Artificial Networks</i>
<b>GBO</b>	<i>Gradient Boosting</i>
<b>GCC</b>	<i>GNU Compiler Collection</i>
<b>GNU</b>	acrônimo recursivo para <i>GNU is Not UNIX!</i>
<b>GP-BO</b>	<i>Gaussian Process - Bayesian Optimization</i>

<b>GRU</b>	<i>Gated-Recurrent Unit</i>
<b>HF</b>	<i>High Frequency</i>
<b>HRV</b>	Variabilidade da Frequência Cardíaca, do inglês <i>Heart Rate Variability</i>
<b>IA</b>	Inteligência Artificial
<b>IAH</b>	Índice de Apneia-Hipopneia
<b>IIR</b>	filtro <i>Infinite Impulse Response</i>
<b>IMC</b>	Índice de Massa Corpórea
<b>IPCA</b>	<i>Incremental Principal Component Analysis</i>
<b>KNN</b>	<i>k-Nearest Neighbor</i>
<b>LB</b>	<i>Lower Bound</i>
<b>LDA</b>	<i>Latent Dirichlet Allocation</i>
<b>LF</b>	<i>Low Frequency</i>
<b>LR</b>	<i>Linear Regression</i>
<b>LSTM</b>	<i>Long Short Term Memory</i>
<b>LightGBM</b>	<i>Light Gradient Boosting</i>
<b>ML</b>	<i>Machine Learning</i>
<b>MLP</b>	<i>Multi Layer Perceptron</i>
<b>MSC</b>	Microsof C
<b>NB</b>	Classificador Naives-Bayes
<b>PCA</b>	<i>Principal Component Analysis</i>
<b>QA</b>	Q-Amplitude
<b>RA</b>	R-Amplitude
<b>REM</b>	<i>Rapid Eyes Moviment</i>
<b>RF</b>	<i>Random Forest</i>
<b>RNN</b>	<i>Recurrent Neuron Networks</i>

<b>ReLU</b>	<i>Rectified Linear Unit</i>
<b>SD</b>	Desvio Padrão
<b>SHAP</b>	<i>Shapley Additive Explanations</i>
<b>SVM</b>	<i>Support Vector Machine</i>
<b>Spec</b>	<i>Specificity</i>
<b>TN</b>	Verdadeiro Negativo, do inglês <i>True Negative</i>
<b>TP</b>	Verdadeiros Positivos, do inglês <i>True Positive</i>
<b>TPE</b>	<i>Tree-structured Parzen Estimator</i>
<b>UB</b>	<i>Upper Bound</i>
<b>UMAP</b>	<i>Uniform Manifold Approximation and Projection</i>
<b>VLF</b>	<i>Very Low Frequency</i>
<b>VM</b>	Voto Majoritário
<b>WoS</b>	<i>Web of Science</i>
<b>XGBoost</b>	<i>eXtreme Gradient Boosting</i>
<b>acc</b>	<i>Accuracy</i>
<b>sen</b>	sensitividade
<b>t-SNE</b>	<i>t-Distributed Stochastic Neighbor Embedding</i>

## LISTA DE SÍMBOLOS

$s(i)$	Coeficiente de silhueta
$T$	Conjunto de treinamento
$M$	Distância entre as margens separadoras da máquina de vetor de suporte
$H_i$	Entropia cruzada
$\phi(\mathbf{x})$	Função <i>kernel</i>
$G_i$	Índice de Gini
$\mathbf{x}$	Matriz ou vetor contendo as características ou espaço de características
$\mathbf{w}$	Matriz ou vetor de pesos
$b$	Matriz ou vetor de vieses
$F1\ score$	Média harmônica entre precisão e recall
$\omega$	Metade do tamanho da janela de dados
$Precision$	Precisão ou taxa de verdadeiros positivos
$Recall$	Sesibilidade, recall ou revocação
$N$	Tamanho total dos dados
$A_{cc}$	Taxa de acerto ou acuracidade
$\alpha$	Taxa de aprendizado

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>29</b>
1.1	JUSTIFICATIVA	29
1.2	OBJETIVOS	30
1.2.1	Objetivo geral	30
1.2.2	Objetivos específicos	31
1.3	METODOLOGIA DE PESQUISA	31
1.4	ESTRUTURAÇÃO DO DOCUMENTO	32
<b>I</b>	<b>REFERENCIAL TEÓRICO</b>	<b>33</b>
<b>2</b>	<b>ARQUITETURA DO SONO E APNEIA OBSTRUTIVA DO SONO</b>	<b>34</b>
2.1	ARQUITETURA DO SONO	34
2.1.1	ESTÁGIOS DO SONO	34
2.1.2	ASPECTOS FISIOLÓGICOS DO SONO	36
2.1.3	DISTÚRBIOS DO SONO	37
2.2	APNEIA OBSTRUTIVA DO SONO	39
2.2.1	CAUSAS, PROBLEMAS RELACIONADOS E DIAGNÓSTICO	39
2.2.2	TRATAMENTOS	41
2.3	RESUMO	42
<b>3</b>	<b>CLASSIFICAÇÃO COM APRENDIZADO DE MÁQUINA</b>	<b>44</b>
3.1	APRENDIZADO DE MÁQUINA	44
3.2	CONCEITOS LIGADOS AO APRENDIZADO	46
3.2.1	Métricas de desempenho	50
3.3	MODELOS DE CLASSIFICAÇÃO DE APRENDIZADO DE MÁQUINAS	52
3.3.1	Classificador naive-Bayes	52
3.3.2	Regressão logística	54
3.3.3	Máquina de vetor de suporte	56
3.3.4	Redes neurais artificiais	59
3.3.5	Modelos baseados em árvore	63
3.3.5.1	Árvores de decisão	64
3.3.5.2	Floresta aleatória	66
3.3.6	<i>Boosting</i>	67
3.3.6.1	<i>Adaptive boosting</i>	67
3.3.6.2	<i>Gradient boosting</i>	68
3.3.7	Modelos do tipo <i>ensemble</i>	71
3.3.7.1	Voto majoritário	72
3.3.7.2	<i>Bagging</i>	73
3.4	REDUÇÃO DE DIMENSIONALIDADE	74
3.4.1	Análise das Componentes Principais	74

3.4.2	<i>t-Stochastic Neighbor Embedding</i> . . . . .	76
3.4.3	<i>Uniform Manifold Approximation and Projection</i> . . . . .	79
3.5	O PROBLEMA DA OTIMIZAÇÃO DOS HIPERPARÂMETROS . . . . .	81
3.5.1	Sintonia dos hiperparâmetros com Optuna . . . . .	82
3.6	EXPLICABILIDADE DOS MODELOS . . . . .	85
3.7	MODELOS DE APRENDIZADO PROFUNDO . . . . .	88
3.7.1	Redes neurais convolucionais . . . . .	89
3.7.2	Redes neurais recorrentes do tipo <i>long-short term memory</i> . . . . .	93
3.8	RESUMO . . . . .	95
<b>4</b>	<b>REVISÃO DA LITERATURA</b> . . . . .	<b>96</b>
4.1	REVISÃO SISTEMÁTICA DA LITERATURA . . . . .	96
4.2	SÍNTESE DAS PRINCIPAIS REFERÊNCIAS . . . . .	106
4.3	RESUMO . . . . .	111

<b>II</b>	<b>MATERIAIS E MÉTODOS</b>	<b>112</b>
<b>5</b>	<b>METODOLOGIA</b> . . . . .	<b>113</b>
5.1	BASE DE DADOS . . . . .	113
5.2	FLUXO DE EXECUÇÃO . . . . .	120
5.3	PRÉ-PROCESSAMENTO E FILTRAGEM DOS ECG . . . . .	122
5.4	EXTRAÇÃO DAS CARACTERÍSTICAS . . . . .	125
5.5	REDUÇÃO DA DIMENSIONALIDADE . . . . .	127
5.6	DIVISÃO DOS DADOS . . . . .	129
5.7	SINTONIA DE HIPERPARÂMETROS . . . . .	129
5.7.1	Modelos de aprendizado de máquinas . . . . .	130
5.7.1.1	<i>Adaptive boosting</i> , AdaBoost . . . . .	130
5.7.1.2	Rede Neural do tipo Multicamadas de Perceptrons . . . . .	131
5.7.1.3	Agregação de Classificadores, <i>Bagging</i> . . . . .	131
5.7.1.4	Árvores de Decisão . . . . .	133
5.7.1.5	<i>Ensemble</i> por Voto Majoritário . . . . .	133
5.7.1.6	<i>Light Gradient Boosting</i> . . . . .	133
5.7.1.7	Regressão Logística . . . . .	134
5.7.1.8	Naive-Bayes . . . . .	135
5.7.1.9	Floresta Aleatória . . . . .	135
5.7.1.10	Máquina de Vetores de Suporte . . . . .	135
5.7.1.11	<i>Extreme gradient boosting</i> , XgBoost . . . . .	136
5.7.2	Modelos de aprendizado profundo . . . . .	136
5.7.2.1	Redes neurais convolucionais, CNN . . . . .	136
5.7.2.2	Redes neurais recorrentes do tipo LSTM . . . . .	137
5.7.2.3	Redes neurais do tipo híbrida entre convolucionais e recorrentes . . . . .	138
5.8	AVALIAÇÃO DE DESEMPENHO E ANÁLISE DE EXPLICABILIDADE . . . . .	139
5.9	EXPERIMENTOS PROPOSTOS . . . . .	140
5.10	RESUMO . . . . .	140

<b>III</b>	<b>RESULTADOS</b>	<b>141</b>
<b>6</b>	<b>RESULTADOS</b>	<b>142</b>
6.1	RESULTADOS DA SINTONIA DAS TÉCNICAS DE REDUÇÃO DE DIMENSÃO	142
6.2	RESULTADOS DOS MODELOS DE APRENDIZADO DE MÁQUINA	143
6.2.1	RESULTADOS GERAIS	144
6.2.2	Resultados de explicabilidade	146
6.2.2.1	Resultados gerais de explicabilidade	146
6.2.2.2	Estudo de caso de explicabilidade: <i>adaptive boosting</i>	149
6.2.3	Resultados da Otimização dos hiperparâmetros e dos modelos	154
6.2.3.1	Máquinas de vetores de suporte (SVM)	154
6.2.3.2	<i>Adaptative Boosting</i> , AdaBoost	158
6.2.3.3	Redes Neurais do tipo multicamadas de <i>perceptrons</i> , MLP	159
6.2.3.4	<i>Bagging</i>	159
6.2.3.5	Árvores de decisão, DT	160
6.2.3.6	<i>Ensemble</i> com voto majoritário, VM	161
6.2.3.7	<i>Light gradient boosting machine</i> , LightGBM	163
6.2.3.8	Regressão logística, LR	164
6.2.3.9	Floresta aleatória, RF	165
6.2.3.10	<i>Extreme gradient boosting</i> , xGBoost	166
6.3	RESULTADOS DOS MODELOS DE APRENDIZADO PROFUNDO	166
6.3.1	RESULTADOS GERAIS	166
6.3.2	Resultados da otimização	168
6.3.2.1	Redes neurais convolucionais, CNN	168
6.3.2.2	Redes neurais recorrentes, LSTM	169
6.3.2.3	Redes neurais híbridas de CNN e LSTM	170
6.4	RESUMO	170
<b>7</b>	<b>DISCUSSÃO</b>	<b>174</b>
7.1	COMPARATIVO GERAL DOS MODELOS	174
7.2	COMPARATIVO DOS EXPERIMENTOS DE DIMENSIONALIDADE	174
7.3	COMPARATIVO COM TRABALHOS SIMILARES	175
7.4	RESUMO	179
<b>IV</b>	<b>CONCLUSÃO</b>	<b>180</b>
<b>8</b>	<b>CONCLUSÃO</b>	<b>181</b>
8.1	OBJETIVOS DA DISSERTAÇÃO	181
8.2	PERSPECTIVAS DE FUTURAS PESQUISA	183
	<b>REFERÊNCIAS</b>	<b>185</b>
	<b>APÊNDICE 1 – RESULTADOS DOS MODELOS APRENDIZADO DE MÁQUINAS E APRENDIZADO PROFUNDO</b>	<b>207</b>

# 1 INTRODUÇÃO

Os distúrbios do sono são responsáveis pela piora da qualidade de vida dos indivíduos, aumento e piora de casos de depressão, considerável piora no desempenho cognitivo e até por acidentes trânsito. Dentre destes distúrbios, a apneia obstrutiva do sono (AOS) tem afetado uma parcela significativa da população, que, além da piora da qualidade do sono e de vida num geral, também tem um risco de agravamento de doenças de insuficiência cardíaca. Muitos dos indivíduos acometidos por AOS não são diagnosticados e acabam não procurando o tratamento devido (SHNEERSON, 2009). O correto diagnóstico da AOS requer a avaliação de diferentes especialistas e da coleta de diversos sinais fisiológicos. Desta forma, estudar, avaliar e propor melhorias ou mesmo novas tecnologias e ferramentas capazes de auxiliar o diagnóstico da AOS, especialmente de forma automática, pode ajudar no diagnóstico e contribuir para que estes indivíduos possam ter acesso ao tratamento adequado (BAHRAMI; FOROUZANFAR, 2022; SHNEERSON, 2009; AYONARA et al., 2014; CARMO et al., 2017).

## 1.1 JUSTIFICATIVA

O ciclo de sono-vigília atua como "relógio interno" do corpo, sendo essencial para diferentes processos fisiológicos. A interrupção do sono pode acarretar em complicações de saúde que afetam a qualidade de vida dos indivíduos. Dos distúrbios do sono, é estimado que a AOS atinja quase 1 bilhão de pessoas em todo o mundo (BENJAFIELD et al., 2019). Estudos mostram que a apneia do sono podem aumentar o risco de doenças cardiovasculares, acidentes vasculares cerebrais (VEASEY; ROSEN, 2019), hipertensão (JAVAHERI et al., 2017), depressão (FAUST et al., 2021), entre outros estados clínicos e podem aumentar o risco do paciente sofrer acidentes de trânsito (AYONARA et al., 2014).

Uma diretriz para o diagnóstico da apneia do sono é o exame de polissonografia, que consiste em coletar diversos sinais biomédicos do paciente em um ambiente clínico. O processo requer a supervisão e análise de especialistas da área da saúde. O paciente precisa dormir fora de seu local habitual enquanto é monitorado por diversos equipamentos e especialistas, tornando o exame caro e inconveniente. Deste modo, estima-se que cerca de 93% das mulheres e 82% dos homens com apneia obstrutiva do sono moderada não são diagnosticados (FAUST et al., 2021; AYONARA et al., 2014).

Em termos de pesquisas recentes, tem sido investigado e desenvolvido técnicas para detecção automática da apneia do sono utilizando menos sinais biomédicos e

menos dependente de especialistas, de forma que uma evidência inicial da apneia do sono seja de mais fácil acesso e menos inconveniente ao paciente. O uso de técnicas de aprendizado de máquinas convencionais (do inglês *machine learning*, ML) e de aprendizado profundo (do inglês *deep learning*, DL) na detecção por meio do eletrocardiograma (ECG) tem se mostrado promissor, tanto pelos resultados obtidos nos estudos, quanto pela possibilidade de detectar apneia do sono com um único sensor, o que poderia viabilizar sua detecção em sensores portáteis ou vestíveis (*wearables* em inglês) (BAHRAMI; FOROUZANFAR, 2022; LI et al., 2018a; YANG et al., 2022; SHETA et al., 2021; MOSTAFA et al., 2019). Modelos de ML e DL são capazes de reconhecer padrões em uma determinada base de dados e generalizar este aprendizado para novos dados, se projetados de forma apropriada (RUSSELL; NORVIG, 2021).

Um dos problemas encontrados são as distintas abordagens empregadas na detecção da AOS. Variando desde dos sinais utilizados, das características extraídas dos dados fisiológicos, dos modelos aplicados e até das bases de dados utilizados (BAHRAMI; FOROUZANFAR, 2022). Toda esta variação pode dificultar a comparação dos resultados obtidos de cada trabalho, pois não apenas a base de dados utilizada é diferente, como alguns trabalhos utilizam o eletroencefalograma (EEG), pressão arterial, níveis de oxigênio e outros sinais, aplicam diferentes tipos de filtros nos sinais e usam formas distintas de sintonizar e testar os modelos de DL e ML (MOSTAFA et al., 2019). Desta forma, criar um fluxo de processamento único, capaz de comparar diferentes características e modelos nas mesmas condições, pode ajudar a entender quais abordagens têm melhores ou piores resultados, como feito em Bahrami e Forouzanfar (2022).

Neste cenário, este trabalho visa contribuir na construção e validação de *frameworks* que buscam auxiliar na detecção automática de AOS através de sinais fisiológicos.

## 1.2 OBJETIVOS

Nesta seção, são listados os objetivos gerais e específicos desta dissertação de mestrado.

### 1.2.1 Objetivo geral

Essa dissertação tem como objetivo geral propor um *framework* que auxilie no treinamento, validação e teste de modelos de inteligência artificial (IA) em diferentes abordagens, gerando, assim, uma base de comparação de modelos e características na detecção de eventos de apneia do sono com uso de sinais de ECG de um único canal, que além de explorar aspectos de desempenho de classificação, também explora

a explicabilidade e a seleção de hiperparâmetros dos modelos. Este *framework* será aplicado na base de dados pública da PhysioNet.

### 1.2.2 Objetivos específicos

De forma a suportar o cumprimento do objetivo geral, essa dissertação possui os seguintes objetivos específicos:

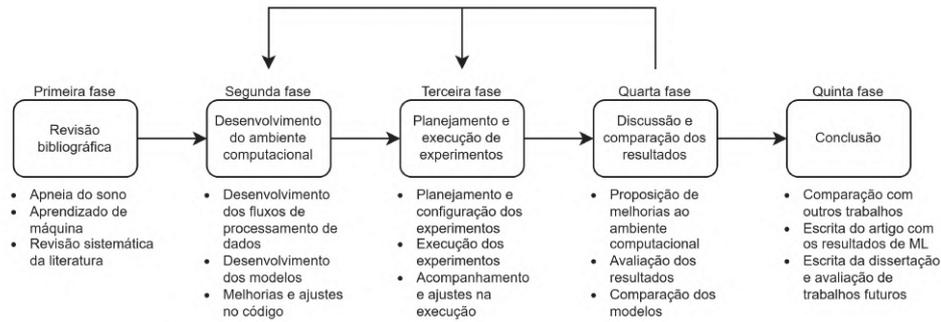
- Realizar pesquisa bibliográfica sobre classificação de apneia do sono utilizando técnicas de aprendizado de máquina e técnicas de aprendizado profundo.
- Aplicar algoritmos para extração das características a partir dos sinais de ECG;
- Avaliar as técnicas de redução da dimensionalidade e estudo das características, tais como análise das componentes principais (do inglês *principal component analysis*, PCA), aproximação e projeção múltipla uniforme (do inglês *uniform manifold approximation and projection*, UMAP) e incorporação estocástica de vizinhança t-distribuída, (do inglês *t-Distributed Stochastic Neighbor Embedding*, t-SNE);
- Conceber a otimização de hiperparâmetros dos classificadores utilizando uma abordagem de otimização Bayesiana;
- Classificar as características extraídas e os sinais processados de ECG quanto a ocorrência ou não de eventos de apneia do sono utilizando modelos de ML e DL;
- Analisar os resultados obtidos dos modelos preditivos utilizando técnicas de desempenho de classificação por meio das seguintes métricas: acurácia (do inglês *accuracy*, Acc), sensibilidade (*recall*, em inglês), precisão e F-1 score;
- Calcular e analisar os valores SHAP (do inglês *Shapley Additive Explanations*), como forma de discutir sobre a explicabilidade dos modelos de ML empregados;
- Comparar os resultados obtidos com artigos de classificação de AOS utilizando técnicas de ML e DL encontrados na literatura recente.

## 1.3 METODOLOGIA DE PESQUISA

Para facilitar a condução da pesquisa e o desenvolvimento do *framework*, um total de cinco fases foram planejadas. A metodologia de pesquisa e fluxo de trabalho, dividido nas fases, são apresentadas na FIGURA 1.

A primeira etapa envolve realizar uma revisão sistemática da literatura a respeito da aplicação de técnicas de ML e de DL na classificação de eventos de AOS

FIGURA 1 – METODOLOGIA DE PESQUISA



FONTE: Próprio autor

LEGENDA: Metodologia de pesquisa e fluxo de trabalho deste projeto

por meio do uso dos sinais de ECG, uma revisão a cerca dos modelos selecionados para estudo e dos temas de redução de dimensionalidade, otimização dos hiperparâmetros, explicabilidade dos modelos de ML e DL e aspectos fisiológicos da AOS. A segunda etapa consiste no desenvolvimento do *framework* e dos modelos e técnicas selecionadas. Com o *framework* desenvolvido, a terceira etapa é planejar e executar os experimentos. Os experimentos variam as técnicas de redução de dimensionalidade empregadas para criar uma base de comparação entre as diferentes abordagens de classificação. Após a execução dos experimentos, a quarta etapa é a análise e discussão dos resultados com base nas métricas de desempenho de classificação (acuracidade, sensibilidade, sensibilidade e F-1 score), em valores de explicabilidade SHAP e medidas de tempo de execução do algoritmo. Tanto a terceira quanto a quarta fase auxiliam na proposição de melhorias ao *framework*, assim como funcionalidades a serem desenvolvidas. Por fim, a quinta fase é a comparação dos resultados obtidos com a literatura e produção de artigos científicos com os estes resultados.

#### 1.4 ESTRUTURAÇÃO DO DOCUMENTO

O restante desta dissertação está estruturado da seguinte forma: os Capítulos 2 e 3 discutem os fundamentos teóricos das técnicas empregadas, o Capítulo 4 expõe uma revisão sistemática da literatura na classificação de eventos da AOS. O Capítulo 5 mostra os materiais e os métodos utilizados na execução do projeto, no desenvolvimento do *framework* e na condução e análise dos experimentos. O Capítulo 6 apresenta os resultados obtidos em cada um dos experimentos propostos. O Capítulo 7 discorre e aprofunda os resultados obtidos, comparando-os com os encontrados na literatura e detalha os achados da pesquisa. Finalizando a dissertação, o Capítulo 8 aborda uma análise dos resultados em termos dos objetivos específicos e do objetivo geral.

## **Parte I**

### **Referencial teórico**

## 2 ARQUITETURA DO SONO E APNEIA OBSTRUTIVA DO SONO

Este capítulo inicia a discussão sobre os fundamentos essenciais ao projeto dessa dissertação. Este capítulo é dividido em duas seções. A primeira seção aborda os estágios, arquitetura e distúrbios do sono. A segunda seção discute os aspectos clínicos da apneia do sono e os tratamentos empregados aos pacientes acometidos com AOS.

### 2.1 ARQUITETURA DO SONO

O sono é definido como estágio fisiológico contrário ao estágio de vigília. O sono é um estágio de inconsciência ao qual o indivíduo pode ser despertado dependendo do tipo estímulo que recebe. Ao contrário do estágio de coma, em que o indivíduo está inconsciente mas não pode ser despertado (SHNEERSON, 2009; GUYTON; HALL, 2006).

O estágio do sono é definido por cinco características básicas: é episódico; é prontamente reversível; existe uma inibição motora; redução da consciência ativa do indivíduo e uma diminuição na capacidade de resposta reduzida (SHNEERSON, 2009). As próximas subseções discutem os estágios, aspectos fisiológicos e distúrbios do sono.

#### 2.1.1 ESTÁGIOS DO SONO

O sono normal em um adulto tem duração de cerca de 7 a 8 horas e pode ser dividido em duas etapas, a fase chamada REM (do inglês *rapid eye movement*) e a fase não-REM. Durante o período de sono, estas duas fases se alternam, realizando cerca de 5 a 6 ciclos completos, com cada ciclo tendo uma duração média de 1 hora. No entanto o estágio REM dura cerca de 5 a 15 minutos por ciclo (PAULA et al., 2020; AYONARA et al., 2014; PRADO et al., 2010). A medida que o sono progredir, o sono não-REM tende a diminuir a duração dos intervalos, enquanto o sono REM tende a ter uma duração maior (SHNEERSON, 2009).

O sono não-REM pode ser dividido em quatro estágios denominados I, II, III e IV. Os estágios I e II caracterizam o denominado sono leve, enquanto os estágios III e IV caracterizam o sono profundo (SHNEERSON, 2009; ROCHA, 2013; PAULA et al., 2020).

O sono leve corresponde a transição do indivíduo do estágio de vigília (acordado) para o estágio de sono. A primeira etapa do sono leve, o estágio I, pode durar de 10 segundos até cerca de 10 minutos. Neste momento é possível perceber uma

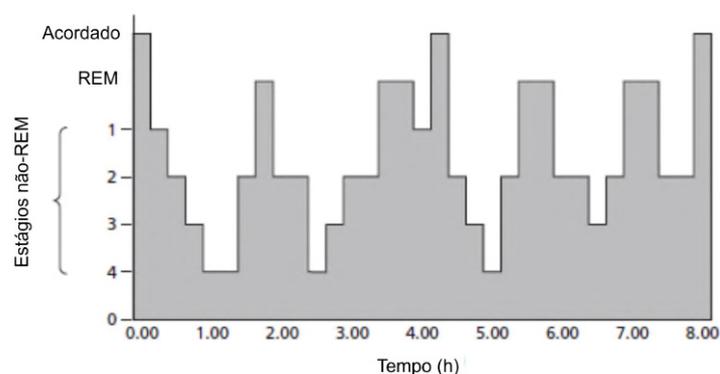
respiração profunda e um relaxamento muscular. A medida que o indivíduo progressivamente continua adormecendo, o estágio II se inicia. Este estágio dura cerca de 10 a 20 minutos e é neste momento que o indivíduo se mantém alheio aos estímulos visuais e sonoros (PAULA et al., 2020).

O sono profundo, também chamado de sono delta ou sono de ondas lentas, é composto pelos estágios III e IV e é caracterizado pela produção de ondas cerebrais de baixa-frequência (PAULA et al., 2020). Nesta etapa do sono, alguns sonhos e pesadelos podem acontecer. A diferença entre estes sonhos para os sonhos que acontecem na fase REM é que os sonhos do sono profundo dificilmente são lembrados, pois não acontece o processo de consolidação em memórias (GUYTON; HALL, 2006).

Após o estágio IV, o sono retorna aos estágios III e II e então entra na fase REM. A primeira fase de sono REM é geralmente curta, durando entre 5 e 15 minutos e a medida que o indivíduo vai descansando com o passar da noite, os episódios de sono REM tendem a aumentar (GUYTON; HALL, 2006; PAULA et al., 2020). As principais características desta fase são: sonhos ativos e movimentos musculares corporais ativos; maior dificuldade de despertar o indivíduo; tônus muscular excessivamente reduzido, o que indica uma forte inibição das áreas de controle de medula espinhal; as ondas do EEG, a frequência cardíaca e a frequência respiratória tendem a ficar irregulares, o que é uma característica dos sonhos; movimentos rápidos oculares e um aumento na atividade do encéfalo, com padrões de EEG semelhantes aos estados de vigília. Devido ao acentuado aumento de atividade cerebral na fase REM, mesmo com o indivíduo dormindo, esta etapa do sono também é conhecida como sono paradoxal (GUYTON; HALL, 2006; ROCHA, 2013).

A FIGURA 2 ilustra a transição entre os estágios de sono, onde o fluxo de transições de estágios é denominado hipnograma (SHNEERSON, 2009).

FIGURA 2 – ESTÁGIOS DO SONO



FONTE: Adaptada de (SHNEERSON, 2009)

LEGENDA: Variação dos estágios de sono de um adulto normal (hipnograma)

### 2.1.2 ASPECTOS FISIOLÓGICOS DO SONO

O sono é fundamental na regulamentação da vida humana. Além de ser responsável por ocupar cerca de um terço do tempo de vida humana, o sono cumpre papéis fundamentais em diferentes aspectos fisiológicos do corpo humano. O sono é considerado como uma fase de descanso do corpo, preparando-o para a próxima etapa de vigília (PAULA et al., 2020; ROPKE et al., 2017; SHNEERSON, 2009).

Alguns processos bioquímicos e fisiológicos acontecem durante o sono. Dois exemplos destes processos são a síntese proteica que é acentuada durante o sono e a secreção do hormônio de crescimento, que ocorre primariamente à noite (SHNEERSON, 2009).

A taxa metabólica do indivíduo diminui durante o sono, com uma diminuição da temperatura corporal. Isto afeta a frequência cardíaca, principalmente durante o sono REM, em que a atividade cardíaca fica instável, devido a atividades físicas. Apesar da pressão arterial não ser controlada pelo ciclo circadiano, existe uma queda nos valores da pressão em cerca de 10 a 15% em relação aos valores no estado de vigília (ROCHA, 2013; SHNEERSON, 2009).

A atividade neuronal varia com a fase do sono. O córtex pré-frontal, responsável pelo planejamento de comportamentos, pensamentos complexos e outras funções, fica inativo durante o sono, tanto na fase não-REM quanto na fase REM. O sono REM pode ter relação com o desenvolvimento neural do indivíduo. Durante esta fase, o córtex é capaz de receber estímulos externos, realizando associações que não são possíveis durante o estado de vigília. Os gânglios basais também estão ativos, desenvolvendo padrões comportamentais sem a necessidade de manifestar em atividade motora. Ainda durante o sono REM, o processamento e integração de informações recém adquiridas em modelos neurais existentes, possibilitando que a experiência adquirida durante o dia possa ser utilizada em ações futuras (SHNEERSON, 2009).

No aspecto psicológico do sono, embora a capacidade de aquisição de novas informações durante o sono seja limitada, o sono possui um papel essencial na consolidação das memórias, com ambas as fases de sono não-REM e REM envolvidas no processo. O aprendizado de informações visuais adquiridas durante o dia acontece principalmente durante os estágios III e IV do sono não-REM, seguido pelo sono REM. O aprendizado das sequências motoras é aprimorado durante o estágio II do sono não-REM. Outras formas de aprendizado, tais como procedimentos cognitivos e associação de eventos também são influenciados pela qualidade dos sonos não-REM e REM. Os sonhos são manifestações das atividades cerebrais e estão associados a processos de ativações de novas redes neurais durante o sono REM. Estas novas redes ajudam na criatividade mental e nas habilidades de resolver problemas (SHNEERSON,

2009).

A função cardiovascular também é afetada durante o sono. Antes mesmo do início da fase não-REM, a frequência cardíaca diminui por conta da diminuição da taxa metabólica, e depois volta a diminuir entre os estágios 2, 3 e 4 da fase não-REM e durante o resto do sono por conta ritmo cardíaco. O sistema cardiovascular é mais estável na fase não-REM do que na REM (SHNEERSON, 2009).

A pressão sanguínea também diminui devido ao ritmo mais lento. O fluxo de sangue para o cérebro durante o sono é menor do que no estado de vigília, sendo que é menor na fase REM do que na fase não-REM. Quando no estado de vigília, em situações de hipóxia, isto é, estados de baixo teor de oxigênio nos tecidos orgânicos devido a diversos fatores, o fluxo sanguíneo cerebral aumenta. Este mesmo mecanismo não é observado durante as fases 3 e 4 do sono não-REM. Como resultado disto, indivíduos acometidos com apneias centrais e obstrutiva podem ter a quantidade de oxigênio disponível no cérebro comprometida (SHNEERSON, 2009).

O sono ainda altera diversos outros mecanismos no corpo humano. Os sistemas digestivos, urinário, imunológico e endócrino são alguns outros exemplos de sistemas com funções alteradas durante os estágios do sono. Tanto sob ponto de vista de fisiológico como sob o ponto de vista de qualidade de vida, um sono de qualidade é crucial para manutenção da saúde (SHNEERSON, 2009). A próxima seção apresenta alguns tipos de distúrbios que podem prejudicar o sono e, com isto, diversos metabolismos corporais.

### 2.1.3 DISTÚRBIOS DO SONO

O sono e o ciclo de sono-vigília podem ser afetados por diferentes fatores e condições. Alguns destes fatores são classificados como situações anormais, como a privação temporária de sono, e por fatores que afetam a natureza em si do sono. Estes fatores são classificados como distúrbios do sono (SHNEERSON, 2009).

Diferentes sintomas podem indicar um distúrbio do sono. De forma simplificada, estes sintomas podem ser agrupados em quatro grandes categorias (SHNEERSON, 2009):

- Insônia, isto é, a dificuldade iniciar e manter continuamente o sono.
- Hipersonia, ou sonolência diurna excessiva, isto é, a dificuldade do indivíduo de se manter acordado durante o período de vigília.
- Comportamentos anormais durante o sono, como por exemplo, sonambulismo e o indivíduo realizar movimentos que acontecem em seus sonhos.

- Distúrbios respiratórios, como por exemplo momentos de falta de ar (apneia) e padrões irregulares de respiração.

Quando um paciente possui reclamações e sintomas de um sono afetado, seja durante o sono em si ou durante a fase de vigília, é necessário realizar uma investigação com objetivo de entender se estes episódios são causados por um distúrbio de sono ou por circunstâncias anormais que estão acometendo o paciente naquele momento. Esta investigação abrange desde avaliar os sintomas em questão, realização de exames fisiológicos, histórico médico, familiar e social, uso ou não de drogas, conversar com possíveis parceiros, entre outros fatores. As causas dos sintomas podem variar, tendo causas mecânicas, incluindo obstrução parcial da entrada de ar, em alguns casos de AOS, neurológicas, que podem afetar os movimentos ou indicar síndrome de Parkinson, psiquiátricas, hormonais e outros. A TABELA 1 apresenta alguns exemplos de sintomas dentro das quatro categorias apresentadas e relaciona estes sintomas com um exemplo de distúrbio de sono que comumente apresenta este sintoma.

**TABELA 1 – EXEMPLOS DE SINTOMAS CAUSADOS POR DISTÚRBIOS DO SONO**

<b>Categoria de sintoma</b>	<b>Exemplo de sintoma</b>	<b>Exemplo de distúrbio do sono</b>
Insônia	Excesso de cansaço  Características de avanço ou atraso de fases do sono	Distúrbio do movimento periódico dos membros Transtorno do sono relacionado com o ritmo circadiano
Hipersônia	Ronco Alteração das características da fase REM	AOS Narcolepsia
Comportamentos anormais	Empurrões e espamos Relacionados a sonhos	Espasmos do sono (mioclonia do sono) Transtorno de estresse pós-traumático
Distúrbios respiratórios	Padrões irregulares de respiração Ronco	Respiração de Cheyne-Stokes  ACS

FONTE: Adaptado de (SHNEERSON, 2009)

Diferentes exames fisiológicos podem ser avaliados para auxiliar o diagnóstico de um distúrbio do sono, sendo a polissonografia um dos mais indicados. A polissonografia consiste em medir simultaneamente e realizar uma análise coordenada em conjunta de diferentes sinais fisiológicos e outros artefatos para estudo, tais como gravação do vídeo e áudio do paciente, durante o sono de forma não-invasiva. Entre os sinais medidos estão o EEG, eletrooculograma, exame que capta a eletrofisiologia ocular, eletromiografia, exame que mede a atividade elétrica dos músculos medidos e ECG. A polissonografia pode ser útil quanto a correlacionar o início e o fim do sono, bem como cada uma das fases do sono, com os sinais fisiológicos, gravação do paciente dormindo, possibilitando um maior número de análises e descobertas. E tem como

desvantagem a fato que o paciente precisa dormir fora de sua residência em um local não habitual, é dependente da análise de um ou mais especialistas, demanda uma grande quantidade de tempo (uma noite dormida) e, devido a estes fatores, possui um custo elevado. Uma vez com o diagnóstico, o especialista clínico pode recomendar o tratamento mais adequado ao caso (SHNEERSON, 2009).

A próxima seção discutirá um dos distúrbios aqui apresentados, a apneia do sono, mais especialmente na versão obstrutiva.

## 2.2 APNEIA OBSTRUTIVA DO SONO

A síndrome da apneia do sono afeta a arquitetura do sono prejudicando as fases do sono, causando recorrentes micro despertares devidos as pausas respiratórias, o que pode acarretar em impactos nas funções neurocognitivas e psicossociais. A apneia pode ser dividida em dois tipos: obstrutiva e central (ACS, apneia central do sono). A AOS acontece devido a uma obstrução temporária da passagem de ar para os pulmões durante o sono, enquanto a ACS, menos frequente na população geral, está relacionada a um distúrbio nos mecanismos de controle da respiração pelo cérebro, medula espinhal e nervos periféricos, prejudicando a respiração (AYONARA et al., 2014; SHNEERSON, 2009; JAVAHERI et al., 2017).

A AOS acomete entre 1-4% de crianças e cerca de 2-4% de adultos, sendo mais predominante em homens (AYONARA et al., 2014). A AOS é um problema de saúde pública e possui etiologia de caráter multifatorial, estando relacionada com idade, sexo, fatores anatômicos e genéticos, gordura corporal, hormônios, entre outros. A AOS pode reduzir significativamente a qualidade de vida dos pacientes e pode agravar o quadro clínico de outras doenças, tal como hipertensão (AYONARA et al., 2014; PEDROSA et al., 2011; PRADO et al., 2010). As próximas seções detalham as causas, problemas relacionados, diagnóstico e tratamento da AOS.

### 2.2.1 CAUSAS, PROBLEMAS RELACIONADOS E DIAGNÓSTICO

Múltiplos fatores podem causar a AOS. Um dos maiores fatores de risco para AOS é a obesidade. A gordura do tecido adiposo que pode se acumular ao redor do pescoço pode obstruir a passagem de ar. Outro fator de risco é o tabagismo, que pode danificar as vias aéreas da boca e da faringe. A posição que o indivíduo dorme também pode ajudar a bloquear parcialmente a passagem de ar. Obstruções nasais, anomalias esqueléticas, a morfologia e danos na região craniofaciais também podem causar AOS. As alterações hormonais, como em casos de acromegalia, hipotireoidismo, diabetes, entre outros, podem afetar a morfologia e os músculos da região da boca e garganta. Distúrbios neurais, tais como a poliomielite por exemplo, que podem afetar os músculos dilatadores das vias aéreas ou que modificam outras regiões da faringe e laringe

podem acarretar em AOS. Finalmente, o uso e consumo de álcool e de medicamentos benzodiazepínicos, como por exemplo alguns tipo de ansiolíticos, podem reduzir o tônus muscular nas vias aéreas contribuindo para ocorrência de AOS (SHNEERSON, 2009).

A AOS pode ser classificada em leve, moderada ou grave, por meio do índice apneia-hipopneia (IAH), de acordo com a TABELA 2. Este índice mede a quantidade de eventos de apneia ou hipopneia dentro de uma hora de sono e é definido na EQUAÇÃO 2.2.1. Um evento de apneia é a obstrução parcial do fluxo de ar com uma redução de pelo menos de 30% do fluxo por pelo menos 10 segundos, este evento é caracterizado pelo aumento do esforço respiratório ou achatamento da pressão nasal do indivíduo (AYONARA et al., 2014; BABY et al., 2013).

$$IAH = \frac{n^{\circ} \text{ de apneias/hipopneias}}{\text{horas de sono}} \quad (2.1)$$

TABELA 2 – USO DO IAH NA CLASSIFICAÇÃO DA SEVERIDADE DA AOS

<b>Classificação</b>	<b>Eventos por hora (IAH)</b>
Leve	6 - 15
Moderada	16 - 30
Grave	>30

FONTE: Adaptada de (AYONARA et al., 2014; BABY et al., 2013)

Diversos problemas estão relacionados a AOS. A fragmentação do sono impacta atenção e foco durante estado de vigília, além de uma piora na função cognitiva, o que pode aumentar risco de acidentes de trânsito e trabalho. Além do cansaço e fadiga, devido a uma piora na qualidade de sono, a AOS pode agravar doenças cardiovasculares, com um aumento associado ao risco de incidência de infartos, insuficiência cardíaca, fibrilação atrial e doença coronariana. A apneia do sono também é a causa secundária mais comum em pacientes com hipertensão arterial resistente. A AOS também está relacionada com distúrbios metabólicos e depressão (JAVAHERI et al., 2017; PEDROSA et al., 2011; AYONARA et al., 2014; ARAGHI et al., 2013).

O padrão-ouro para diagnóstico é a polissonografia, sendo o exame apropriado para o cálculo do IAH, podendo ser auxiliado pela análise do histórico clínico e por outros exames, a citar por exemplo a oximetria noturna, avaliações físicas e a cefalometria. No histórico clínico é avaliado se existem quadros de sonolência excessiva durante a fase de vigília ou se existem relatos de ronco ou pausas respiratórias pelo companheiro, sendo estes os principais sintomas. O exame físico precisa avaliar o peso e o índice de

massa corpórea (IMC) do indivíduo, outras medidas como a circunferência do pescoço, por exemplo, também pode ajudar no diagnóstico. O exame da cavidade nasal pode avaliar possíveis obstruções nasais, como por exemplo desvio de septo. A cefalometria atua como exame diagnóstico complementar, neste exame é medido o diâmetro das vias aéreas, fornecendo informações importantes sobre a relação maxilo-mandibular. A oximetria noturna mede a saturação de oxigênio durante a noite, podendo ajudar a identificar instabilidades na respiração e auxiliando o diagnóstico (PRADO et al., 2010; AYONARA et al., 2014).

O diagnóstico da polissonografia pode ser conclusivo e ajuda a classificar a gravidade da AOS, no entanto, para o tratamento pode ser necessário realizar investigação multidisciplinar para melhor determinar as causas (PRADO et al., 2010). A próxima seção aborda algumas opções de tratamento para AOS.

### 2.2.2 TRATAMENTOS

O tratamento da AOS é multidisciplinar e pode envolver diferentes profissionais da área da saúde e diversas especialidades médicas. Pediatras, otorrinolaringologistas, pneumologistas, fonoaudiólogo, cirurgião-dentista, fisioterapeuta, entre outros desempenham papel importante no diagnóstico e tratamento dos pacientes. Existem diferentes métodos de tratamento, que vão desde alterações comportamentais, uso de dispositivos orais e de aparelhos de pressão aérea superior positiva contínua (CPAP, do inglês *continuous positive airway pressure*, intervenções cirúrgicas, uso de fármacos e até terapia miofuncional (AYONARA et al., 2014; PRADO et al., 2010).

Tratar as causas da AOS requer educar o paciente a respeito da condição e pode ser necessário testar mais de um tratamento, a depender da causa. Mudanças de comportamentos e hábitos podem ajudar a reduzir o IAH, desde da mudança da postura ao dormir e até adoção de hábitos mais saudáveis como exercícios físicos, redução de peso, evitar consumo de álcool e de certas drogas. O tratamento mais utilizado é o uso de aparelhos CPAP, estes dispositivos geram um fluxo contínuo de ar através de um tubo flexível conectado a uma máscara nasal que é colocada no rosto do paciente, como mostra a FIGURA 3. Estudos mostram que o CPAP diminuiu a apneia do sono em aproximadamente 50% dos pacientes. Dispositivos orais também podem ser utilizados, como por exemplo aparelhos de avanço mandibular e de retenção de língua, promovendo alterações anatômica nas estruturas das vias aéreas. Um exemplo de aparelho de avanço mandibular é mostrado na FIGURA 4. Pacientes com anormalidades anatômicas podem ser submetidos a intervenção cirúrgica. A cirurgia tem como objetivo corrigir a obstrução do fluxo de ar. O tratamento farmacológico não é muito utilizado e consiste em no uso de esteroides nasais para pacientes com AOS e com rinite ou usar um medicamento que diminua a sonolência durante o dia, quando

outras causas da sonolência também foram detectadas. Um estudo revelou que a terapia miofuncional pode ajudar a corrigir musculatura das vias aéreas e diminuir o IAH (AYONARA et al., 2014; BABY et al., 2013).

Os tratamentos podem ajudar a melhorar a qualidade de vida do paciente. Para isso é crucial o correto diagnóstico da AOS. As próximas seções vão explorar técnicas de modelagem de ML que podem ser utilizadas com objetivo de auxiliar a detecção da AOS.

FIGURA 3 – EXEMPLO DE APARELHO CPAP



FONTE: (ANÔNIMO, s.d.[b])

LEGENDA: Exemplo de um aparelho CPAP

FIGURA 4 – EXEMPLO DE APARELHO ORAL DE AVANÇO MANDIBULAR



FONTE: (ABI-RAMIA et al., 2010)

LEGENDA: Exemplo de um aparelho oral de avanço mandibular

### 2.3 RESUMO

O sono é essencial na manutenção da saúde do indivíduo. Desta forma, este capítulo apresentou os estágios do sono e a função de cada um desses estágios no desenvolvimento e manutenção do corpo humano. Também foram discutidos os distúrbios do sono, com enfoque na AOS, detalhando os aspectos clínicos e opções de tratamentos para pacientes acometidos. O próximo capítulo continua a discussão dos

fundamentos essenciais ao projeto, abordando a modelagem com uso de aprendizado de máquinas.

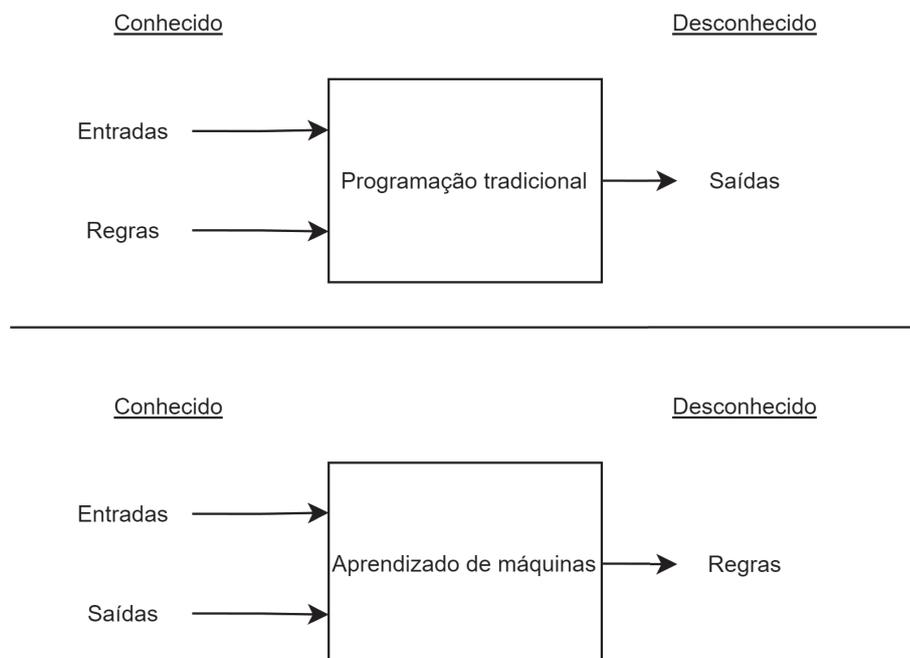
### 3 CLASSIFICAÇÃO COM APRENDIZADO DE MÁQUINA

Este capítulo aborda os fundamentos de modelagem com ML, das etapas necessárias, as técnicas utilizadas e avaliação dos resultados. A primeira seção do capítulo introduz o que é ML e o que é necessário aplicar quando se implementa uma modelagem com ML.

#### 3.1 APRENDIZADO DE MÁQUINA

Aprendizado de máquina pode ser definido como conjuntos de técnicas de modelagem que permitem extrair regras e realizar inferências, por meio de um conjunto de dados, sem serem explicitamente programados para esta função. As técnicas de aprendizado de máquina são utilizadas em problemas em que os padrões dos dados não são totalmente conhecidos ou não seja possível codificá-los totalmente (GÉRON, 2019). A FIGURA 5 exibe um comparativo entre uma abordagem de programação dita tradicional e uma abordagem por aprendizado de máquina. O aprendizado de máquina é um campo de estudo da inteligência artificial (RAMACHANDRAN; KARUPPIAH, 2021; RUSSELL; NORVIG, 2021).

FIGURA 5 – COMPARATIVO ENTRE PROGRAMAÇÃO TRADICIONAL E APRENDIZADO DE MÁQUINA



FONTE: Próprio autor

LEGENDA: Comparação entre abordagem da programação tradicional com aprendizado de máquina

As técnicas de aprendizado de máquinas são comumente empregadas em cinco tipos de problemas: classificação, em que o objetivo é inferir a classe de uma nova entrada de dado com base em seus atributos, dentro de um conjunto finito de classes possíveis; regressão, que diferente do problema de classificação, a inferência é feita em um valor contínuo; recomendação, cujo o principal objetivo é sugerir uma ação, objeto ou produto similar; otimização, em que o resultado é um conjunto, combinação ou valor que otimizam o problema e criação de dados sintéticos, em busca adicionar novos dados artificiais dentro de uma base de dados. As técnicas também podem ser divididas em quatro tipos (DAS; BEHERA, 2017): aprendizado supervisionado, em que o processo de aprendizagem ocorre com dados rotulados; aprendizado não supervisionado, em que os dados apresentado ao modelo não possuem rótulos; aprendizado semi-supervisionado, em que parte dos dados possuem rótulos e parte não possui e aprendizado por reforço, em que existe um método de recompensa para as melhores soluções encontradas pelo algoritmo (HASTIE et al., 2009b; JORDAN; MITCHELL, 2015; CERRI; LEON FERREIRA et al., 2019).

Nesta dissertação, foram utilizadas técnicas de classificação para identificar eventos de apneia. Dentre as diversas técnicas de classificação existentes, forma aplicadas neste trabalho:

- De ML:
  - Árvores de decisão (DT, do inglês *decision trees*)
  - Floresta aleatória (RF, do inglês *random forest*)
  - Máquinas de vetores de suporte (SVM, do inglês *support vector machines*)
  - Classificador Bayesiano (Nayves-Bayes, NB)
  - Redes neurais artificiais do tipo multicamadas de perceptrons (MLP, do inglês *multi-layer perceptron*)
  - Regressão logística (LR, do inglês *linear regression*)
  - Aglomeração de classificadores (*bagging*)
  - *Light gradient boosting machine* (LightGBM)
  - *eXtreme gradient boosting machine* (XgBoost)
  - *Adaptive boosting* (AdaBoost)
  - *Ensemble* por voto majoritário (VM)
- De DL:
  - Redes neurais convolucionais (CNNs, do inglês *convolutional neural network*)

- Redes neurais de memória de longo prazo (LSTM, do inglês *long-short term memory*)
- Redes neurais de arquitetura híbrida, com camadas convolucionais e recorrentes (referenciadas como híbridas de CNN e LSTM)

As próximas seções abordam conceitos e detalham estes modelos e outros métodos dentro do processo de modelagem com ML e DL. A seção 3.2 aborda conceitos que se aplicam a classificação com uso de aprendizado de máquinas e a todas as técnicas descritas de ML e DL. A seção 3.3 apresenta os modelos de ML com detalhes, em seguida as seções 3.4, 3.5.1 e 3.6 apresentam os fundamentos desse processo de modelagem, detalhando algumas técnicas utilizadas para redução de dimensionalidade, seleção de hiperparâmetros e explicabilidade dos resultados, respectivamente. Por fim a subseção 3.7 introduz e aborda os detalhes dos modelos de DL.

## 3.2 CONCEITOS LIGADOS AO APRENDIZADO

Aprendizado pode ser definido como o ato de um determinado agente melhorar o seu desempenho a partir de observações que o mesmo fez do mundo em que se encontra. No caso de aprendizado de máquina, o agente em questão é um computador. O computador observa alguns dados e os utiliza como base para a construção de um modelo, isto é, um objeto computacional que implementa uma ou algumas equações e interligações entre as mesmas. Este modelo é utilizado tanto como uma hipótese sobre este mundo como um *software* capaz de solucionar problemas (RUSSELL; NORVIG, 2021).

Três fatores fundamentais impactam a melhora de desempenho de um modelo. O primeiro fator é o tipo de componente a ser melhorado; o segundo é o conhecimento já existente (*a priori*) a cerca do problema e o terceiro fator são os dados disponíveis e o tipo de resposta (*feedback*) que é possível perceber ou medir a cerca de uma determinada ação do agente. Existem três principais tipos de resposta que determinam o tipo de aprendizado (MUHAMMAD; YAN, 2015; JORDAN; MITCHELL, 2015):

- No aprendizado supervisionado, o agente dispõe de sinais de entrada e saída correspondentes, de modo que é possível construir uma função que mapeia aquela entrada para aquela saída (chamada de rótulo ou *label*). Esta função pode ser adaptada depois para criar uma generalização em cima dos padrões aprendidos. Em outras palavras, o modelo aprende a partir de exemplos rotulados e então tenta generalizar este aprendizado para novos exemplos.
- Diferentemente do que ocorre no aprendizado não-supervisionado, em que o modelo não dispõe de um conjunto de dados de saída rotulados. A partir apenas

do conjunto de entrada, o modelo cria uma função que busca identificar padrões nos dados. Uma aplicação muito comum de aprendizado não-supervisionado é o agrupamento, em que objetivo é agrupar elementos que sejam similares em alguns aspectos.

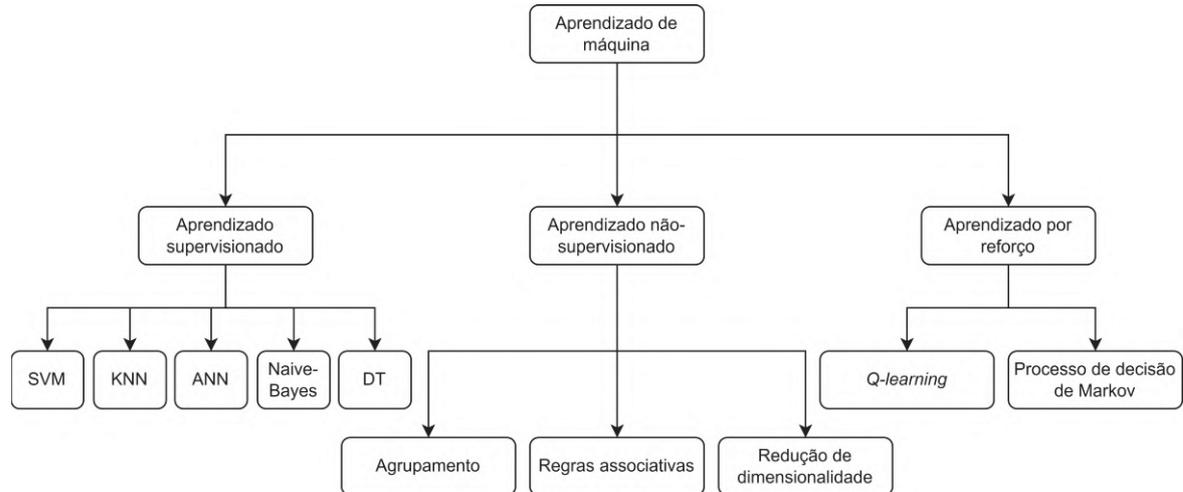
- O aprendizado semi-supervisionado combina ambos os aspectos supervisionado e não-supervisionado. Parte dos dados, normalmente a menor parte, possuem rótulos confiáveis atribuídos ou revisados por um humano e parte dos dados, geralmente a maior parte dos dados, não possui rótulos ou os rótulos não são confiáveis. Isto acontece pois o processo de rotular os dados (*labeling* em inglês) pode ser custoso e caro, enquanto pode ser mais fácil capturar os dados sem rótulos. Existem diferentes abordagens para este tipo problema, como por exemplo agrupar os dados e utilizar os rótulos disponíveis para atribuir aos dados.
- No aprendizado por reforço, no aprendizado existe um mecanismo de *feedback* que recompensa ou pune o agente de acordo com as ações tomadas em um objetivo específico. Com base nestes *feedbacks*, o agente aprende e reforçar as melhores ações.

A FIGURA 6 ilustra alguns exemplos de tipos de modelos, técnicas e problemas abordados dentro de cada uma destas categorias de aprendizado. Na FIGURA 6, alguns exemplos de modelos de aprendizados supervisionados são mostrados, SVM, ANN, KNN (do inglês *k neaghest neighbors*, k vizinhos próximos), naive-Bayes e DT. Dentro de aprendizado não-supervisionado, são ilustrados problemas comumente associados, como agrupamento de dados, modelagem por regras associativas e redução de dimensionalidade. Enquanto *Q-learning* e processo de decisão de Markov são exemplos de técnicas de aprendizado por reforço.

O processo de aprendizado do modelo é chamado de treinamento e consiste em configurar os pesos de cada equação e a topologia que compõe o modelo em questão de acordo com os dados imputados. Para um modelo de aprendizado supervisionado isto é feito utilizando diferentes algoritmos, à medida que exemplos rotulados são apresentados para o modelo e o resultado é avaliado (GÉRON, 2019).

Como os dados do problema podem assumir qualquer formato, estruturado (numérico, data ou categórico) ou não-estruturado (textos, imagens, áudios, entre outros) é necessário preparar os dados para treinamento. Pode ser necessário tratar algum tipo de anomalia ou de estrutura nos dados, como por exemplo: dados incompletos, remoção de acentos em palavras e textos, transformações de tipos, correções de ponto flutuante, entre outros tratamentos. Caso exista dados categóricos, textos, imagens e outros formatos não numéricos, será necessário aplicar alguma transformação e transforma-los em numéricos, pois a grande maioria das técnicas de ML apenas traba-

FIGURA 6 – TIPOS DE APRENDIZADO COM EXEMPLOS DE TÉCNICAS E PROBLEMAS APLICÁVEIS



FONTE: Fonte: o autor, adaptada de (LAKSHMANAN et al., 2020)

LEGENDA: Tipos de aprendizado com exemplos de algumas técnicas e modelos e problemas aplicáveis

lham com dados numéricos (MUHAMMAD; YAN, 2015; QIU et al., 2016; LAKSHMANAN et al., 2020).

Novas informações também podem ser extraídas de alguns tipos de dados, como por exemplo, extrair o dia da semana de um formato de data, ou a contagem de quantas vezes uma palavra aparece no conjunto de dados (QIU et al., 2016; LAKSHMANAN et al., 2020).

Todo este preparo dos dados é comumente referido como etapa de pré-processamento dos dados. Após o pré-processamento, os dados passam a ser chamados de características (ou *features*). Estas características são, então, divididas em três conjuntos: treinamento, validação e teste (LAKSHMANAN et al., 2020; QIU et al., 2016).

O conjunto de treinamento será utilizado para que o modelo calibre os pesos e topologias e tente encontrar padrões ali contidos. O conjunto de validação é também utilizado no processo de treinamento, para que o modelo possa avaliar o desempenho e recalibrar os pesos e topologias. Por fim, o conjunto de teste tem papel de medir a capacidade do modelo de generalização, uma vez que o modelo não foi apresentado a este conjunto de dados previamente (JORDAN; MITCHELL, 2015; CERRI; LEON FERREIRA et al., 2019).

Em alguns tipos de modelos, certos tipos de hiperparâmetros, que são configurados antes do treinamento, vão conduzir o modo como os pesos e a topologia serão ajustados. É comum que estes hiperparâmetros também passem por uma etapa de busca, alternado a busca destes hiperparâmetros com treinamento dos modelos. Avaliando-se assim, as melhores configurações de hiperparâmetros para aquele

problema (LAKSHMANAN et al., 2020).

Para um treinamento adequado, é necessário garantir que os três conjuntos de dados - treinamento, validação e teste - possuam propriedades estatísticas semelhantes. Para o tipo de aprendizado supervisionado, é importante garantir um balanceamento nos rótulos alvo. Além disso, a qualidade dos dados cumpre um papel fundamental para um ótimo desempenho do modelo. Uma vez que todo o processo de aprendizado é pelos os dados, os resultados dependerão do quão bem aqueles dados representam o problema em específico. Desde de exemplos suficientemente diversificados, de modo que se diminua o viés da base de dados, até rótulos que sejam realmente a verdade absoluta (ou fundamental). Durante a implementação de uma solução de ML, a qualidade dos dados é analisada repetidamente, afim de garantir uma base coerente (PALEYES et al., 2022; QIU et al., 2016).

Durante o treinamento, o modelo passará por um processo de busca de quais os melhores pesos melhor se ajustam ao problema, utilizando a base de validação para medir seu desempenho. As saídas do modelo são chamadas de previsões ou inferências. Dentro deste contexto acontece o chamado compromisso viés-variância (em inglês *bias-variance tradeoff*) (PALEYES et al., 2022).

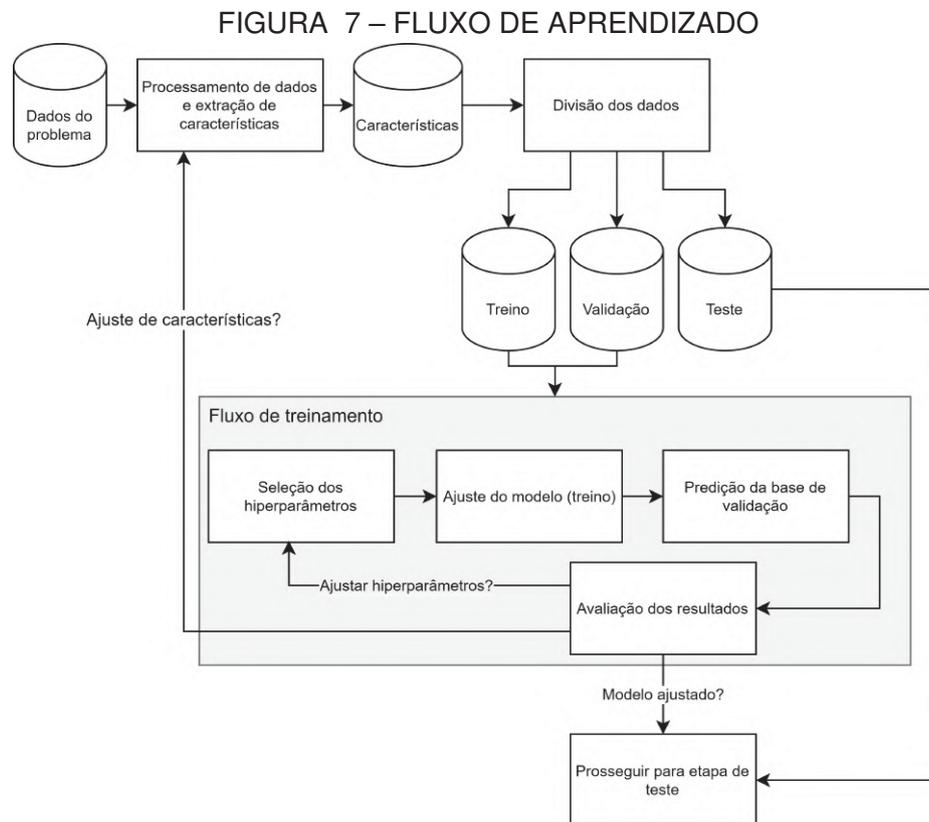
O viés é definido, neste contexto, como a tendência do das previsões divergirem do resultado do valor esperado quando aplicado a diferentes conjuntos de treinamento. O viés pode muitas vezes acontecer devido as escolhas dos valores dos hiperparâmetros e do tipo de modelo (QIU et al., 2016).

A variância, por sua vez, é definida neste contexto pela amplitude da mudança nos pesos e forma do modelo para uma mudança no conjunto de treinamento. Se uma mudança pequena no conjunto de treinamento causa uma pequena mudança no modelo, o modelo possui uma baixa variância. Caso contrário, o modelo possui uma alta variância (CERRI; LEON FERREIRA et al., 2019).

O compromisso viés-variância é a escolha de um modelo mais complexo com menor viés que se ajusta bem aos dados de treinamento ou um modelo mais simples, de baixa variância que pode ser capaz de generalizar melhor. Esta escolha é crucial para garantir que não aconteça subajustamento (*underfitting*) ou sobreajustamento (*overfitting*) do modelo. O subajustamento é quando o modelo não é capaz de reconhecer nenhum padrão nos dados e apresenta um desempenho ruim, enquanto o sobreajustamento acontece quando o modelo se adapta tão bem aos dados de treinamento que se torna incapaz de generalizar o aprendizado, apresentando uma elevada taxa de erro quando novos exemplos são imputados (MUHAMMAD; YAN, 2015).

Na FIGURA 7 É exibido um resumo do processo de aprendizado, que segue as etapas: processamento dos dados para geração das características, divisão da base

de dados em treino, validação e teste, seleção dos hiperparâmetros, ajuste do modelo e avaliação dos resultados.



FONTE: Fonte: o autor

LEGENDA: Fluxo de aprendizado

### 3.2.1 Métricas de desempenho

As métricas de desempenho podem variar de acordo com o problema em questão, tanto na natureza do tipo de métrica, quanto o que é considerado um bom desempenho. Esta discussão é extremamente importante, pois a métrica de desempenho vai direcionar o desenvolvimento do modelo e os avanços do projeto para fases seguintes, como teste do modelo e até sua implementação (RUSSELL; NORVIG, 2021; LAKSHMANAN et al., 2020; HASTIE et al., 2009b; GÉRON, 2019).

Neste projeto, é abordado um problema de classificação binário, isto é, com duas classes. Com base nisto, a discussão das métricas será centrada neste tipo de problema, muito embora parte da discussão se estenda a problemas multiclases, para maiores informações o leitor pode consultar Russell e Norvig (2021), Géron (2019) e Hastie et al. (2009b).

Quando a classificação possui apenas duas classes, chamadas de positiva (abreviada para P) e negativa (abreviada para N), diferentes métricas podem ser extraídas e avaliar o quão bom foi o desempenho geral e o desempenho particular em

cada uma das classes. Dentro do contexto desta dissertação, a classe positiva são os eventos de AOS, enquanto a classe negativa são os eventos de não-AOS (CARVALHO et al., 2019).

Uma predição é dita como verdadeira positiva (TP, do inglês *true positive*), se o modelo atribuiu a classe positiva e o rótulo original é positivo. De forma análoga, é dita verdadeira negativa (TN, do inglês *true negative*) quando é predito e o rótulo original é negativo. Esta ideia se estende para falso positivo (FP, do inglês *false positive*), quando o modelo atribui um rótulo positivo mas a classificação original é negativa. Este tipo de erro é chamado de erro tipo I, quando é atribuído uma condição para um elemento, mas esta condição não existe (HOSSIN; SULAIMAN, 2015).

Similarmente, falso negativo (FN, do inglês *false negative*) acontece quando o resultado atribuído é negativo, mas o rótulo original é positivo. Este é o erro tipo II, quando existe uma condição no elemento, mas o resultado mostra erroneamente que não (HOSSIN; SULAIMAN, 2015).

A partir destes conceitos, é definida a matriz de confusão, conforme mostrado na FIGURA 8. Na FIGURA, as colunas indicam os resultados preditos, enquanto as linhas os resultados reais. Desta forma, a diagonal (em verde) representa os acertos do modelo, e as demais células os erros do modelo (em vermelho na FIGURA). A célula em cinza, representa o total de predições feitas pelo modelo (HOSSIN; SULAIMAN, 2015; HASTIE et al., 2009b).

FIGURA 8 – MATRIZ DE CONFUSÃO

		Classes preditas	
		Positivos (P)	Negativos (N)
Classes reais	Total = P + N		
	Positivos (P)	Verdadeiro Positivo (TP)	Falso Negativo (FN)
	Negativos (N)	Falso Positivo (FP)	Verdadeiro Negativo (TN)

FONTE: Fonte: o autor

LEGENDA: Matriz de confusão

A partir da matriz de confusão, pode-se definir a acuracidade ou taxa de acerto ( $A_{cc}$ ) como uma métrica que mede o desempenho geral do modelo, isto é a proporção de predições corretas em relação ao total de predições feitas pelo modelo, definidas como (HOSSIN; SULAIMAN, 2015):

$$Acc = \frac{(TP + TN)}{TP + FP + TN + FN} \quad (3.1)$$

A precisão (ou *Precision* em inglês) como uma métrica que mede dentro da classe positiva, quanto o modelo acertou de positivo:

$$Precision = \frac{TP}{TP + FP} \quad (3.2)$$

A sensibilidade, *Recall* ou revocação mede quantas vezes o modelo acertou a classe positiva dentro todas as vezes que a classe positiva era o valor esperado (HASTIE et al., 2009b):

$$Recall = \frac{TP}{TP + FN} \quad (3.3)$$

Por fim, o *F1 score* é a média harmônica entre precisão e *recall* (HASTIE et al., 2009b):

$$F1 \text{ Score} = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \quad (3.4)$$

Em geral, modelos de classificação são calibrados em torno de uma dessas métricas. Acuracidade é muito utilizada quando a assertividade geral do modelo é importante. A precisão é utilizada em contextos que consideram os FP mais prejudiciais que os FN. *Recall* condensa duas métricas em uma e pode ser útil no contexto que ambos FN e FP são prejudiciais. Neste projeto, a acuracidade foi utilizada como métrica principal, por se tratar de uma medição mais geral do desempenho (HOSSIN; SULAIMAN, 2015).

A próxima seção comenta quanto aos detalhes dos modelos de aprendizado de máquina.

### 3.3 MODELOS DE CLASSIFICAÇÃO DE APRENDIZADO DE MÁQUINAS

Nesta seção são descritos os modelos de ML ditos clássicos, enquanto na seção 3.7 são descritos os modelos de DL. Para facilitar a leitura, todos os modelos que se baseiam na construção de árvores foram descritos na subseção 3.3.5.1 e os modelos que combinam diferentes tipos de modelos (*ensemble*) são descritos na subseção 3.3.7.

#### 3.3.1 Classificador naive-Bayes

O classificador Bayesiano ou naive-Bayes aplica o Teorema de Bayes para inferir a classe mais provável de um determinado elemento. O cálculo desta probabilidade

é feito a partir do conjunto de dados histórico utilizados para o treinamento do modelo, atribuindo-se pesos a cada uma das probabilidades individuais. O processo de aprendizado do modelo é basicamente uma inferência probabilística (WICKRAMASINGHE; KALUTARAGE, 2021).

O modelo atribui probabilidades para cada um dos elementos e para cada classe. Seja  $\mathbf{x}$  o vetor que representa todos os dados (características) e  $x_i$  um valor observado deste conjunto de um total de  $N$  observações, e  $K$  possibilidades de classe sendo  $C_k$  uma classificação atribuída a um elemento. Na abordagem Bayesiana, é calculado a probabilidade *a priori*,  $P(C_k)$ , a verossimilhança de uma instância do dado pertencer a uma classe específica,  $P(C_k|\mathbf{x})$  e a probabilidade daquela instância de dados  $P(\mathbf{x})$ . Pelo teorema de Bayes, a probabilidade de cada dado pertencer a uma classe é dada por (KAUR; OBERAI, 2014):

$$P(C_k|\mathbf{x}) = \frac{P(\mathbf{x}|C_k)P(C_k)}{P(\mathbf{x})} \quad (3.5)$$

Na prática, apenas o numerador importa para realização de uma inferência, que corresponde à probabilidade conjunta dos dados com a classe. Considerando que todos os elementos em  $\mathbf{x}$  são mutualmente independentes e aplicando a regra de probabilidade condicional, obtém-se que (WICKRAMASINGHE; KALUTARAGE, 2021; KAUR; OBERAI, 2014; RUSSELL; NORVIG, 2021):

$$P(C_k|x_1, \dots, x_N) = \alpha P(C_k) \prod_{i=1}^n P(x_i|C_k) \quad (3.6)$$

onde  $\alpha$  é uma constante de proporcionalidade, uma vez que o denominador com o termo  $P(\mathbf{x})$  foi omitido. Logo, com a condição dos dados serem mutualmente independentes, a distribuição de probabilidades condicionais da class  $C$  é dada por (WICKRAMASINGHE; KALUTARAGE, 2021; KAUR; OBERAI, 2014; RUSSELL; NORVIG, 2021):

$$P(C_k|x_1, \dots, x_n) = \frac{1}{Z} P(C_k) \prod_{i=1}^n P(x_i|C_k) \quad (3.7)$$

onde  $Z = P(\mathbf{x}) = \sum_K P(C_k)P(\mathbf{x}|C_k)$ . Se todos os valores das características são conhecidos, então  $Z$  é uma constante. O classificador Naive-Bayes, então, combina

este modelo probabilístico com uma regra de decisão, que é, dentro das  $K$  classes disponíveis, qual é a classe que maximiza a probabilidade a *posteriori* para aquela instância de dados. Portanto o classificador Naive-Bayes é dada por (WICKRAMASINGHE; KALUTARAGE, 2021; KAUR; OBERAI, 2014; RUSSELL; NORVIG, 2021):

$$\hat{y} = \arg \max_{k \in \{1, \dots, K\}} P(C_k) \prod_{i=1}^n P(x_i | C_k) \quad (3.8)$$

onde  $\hat{y} = C_k$ , atribui o rótulo da classe.

Dos modelos aplicados neste projeto, o classificador Naive-Bayes é o único que não possui nenhum hiperparâmetro a ser configurado antes do treinamento. Isto acontece pois todo o modelo é construído com base nas distribuições dos dados e classes do problema (REDDY et al., 2022). Exemplos de aplicação do classificador Naive-Bayes são diversas e muitas vezes combinam outras técnicas, alguns exemplos são detecção de fraudes no resgate de seguros (VIAENE et al., 2004), prever interação de proteínas (MURAKAMI; MIZUGUCHI, 2010) e predição de geração de energia fotovoltaica (BAYINDIR et al., 2017).

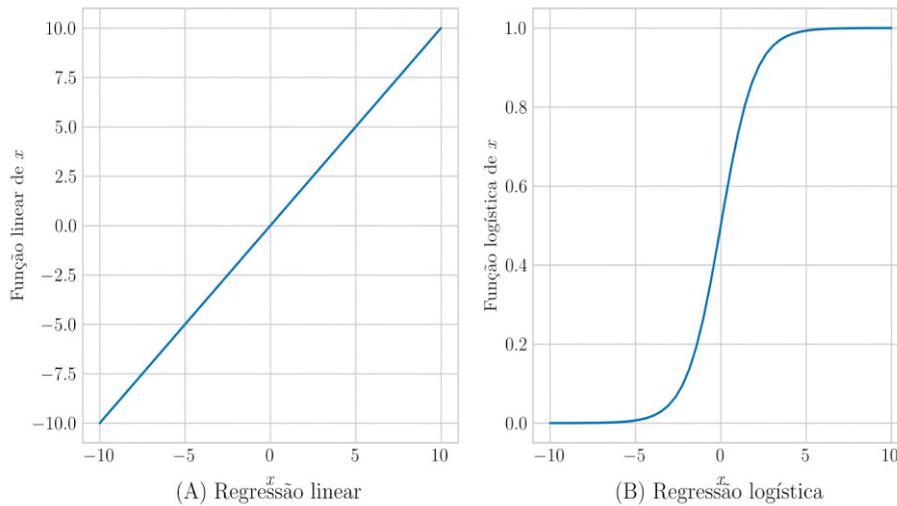
### 3.3.2 Regressão logística

A regressão logística é um modelo similar à regressão linear, porém utilizada para classificação. Enquanto a regressão linear aplica o método dos mínimos quadrados para minimizar o erro dos resíduos e obtém uma reta capaz de prever um valor, a regressão logística aplica uma maximização da verossimilhança, de forma a obter os valores máximos de probabilidade dos dados observados em uma curva logística (também chamada de sigmoide ou curva em forma de s). Por se tratar de uma curva logística, a saída do modelo é um valor de probabilidade que é convertido para 0 (se menor que 50%) ou 1 (se maior que 50%). A FIGURA 9 ilustra a forma de uma curva de regressão linear e logística assumem. Na FIGURA 9, a curva exemplo "A" na esquerda, corresponde a curva de regressão linear, enquanto a curva "B" na direita corresponde a curva de regressão logística (LAVALLEY, 2008; NASTESKI, 2017). Para uma matriz de treinamento  $\mathbf{x}$ , em que cada coluna é uma característica diferente, a equação 3.9 define a regressão logística. Na equação,  $\mathbf{w}$  é um vetor de pesos que é ajustado durante o processo de treinamento (RUSSELL; NORVIG, 2021).

$$\text{Logística}(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w} \cdot \mathbf{x}}} \quad (3.9)$$

Diferente da regressão linear, em que o método dos mínimos quadrados é capaz de encontrar a solução ótima, na regressão logística, uma das formas de encontrar os valores ótimos dos pesos é aplicar o método de descida de encosta. Para

FIGURA 9 – COMPARATIVO ENTRE AS CRUVAS DE REGRESSÃO LINEAR E LOGÍSTICA



FONTE: Fonte: o autor

LEGENDA: Exemplo de curvas de regressões linear e logística

um problema de classificação binária e para um exemplo particular simples  $(x, y)$ , em que  $y$  é o rótulo daquele exemplo, assumindo um valor de 0 ou 1. Se definirmos a função  $Loss(\mathbf{w})$  a função de perda que calcula o erro de classificação dado um vetor de ponderações, a derivação do gradiente descendente para este exemplo é mostrado na equação 3.10 (BEWICK et al., 2005; MAALOUF, 2011):

$$\frac{\partial}{\partial w_i} Loss(\mathbf{w}) = \frac{\partial}{\partial w_i} (y - \text{Logística}(x))^2 \quad (3.10)$$

É possível demonstrar, que a partir da equação 3.10, a atualização do peso  $w_i$  é dada pela equação 3.11 (BEWICK et al., 2005; MAALOUF, 2011):

$$w_i = w_i + \alpha (y - \text{Logística}(x)) \times \text{Logística}(x) (1 - \text{Logística}(x)) \times x_i \quad (3.11)$$

Em que  $x_i$  é uma instância da matriz de treinamento e  $\alpha$  é um hiperparâmetro conhecido como taxa de aprendizado, que possibilita que as atualizações de  $w_i$  sejam suaves, de forma que evite os valores de  $w_i$  sofram variações abruptas, evitando *underfitting* e *overfitting*. O processo de treinamento é repetido aplicando a equação 3.11, por uma quantidade pré-definida de iterações seja atingida ou a matriz de pesos fique estável e não sofra mais alterações (LAVALLEY, 2008; BEWICK et al., 2005; MAALOUF, 2011).

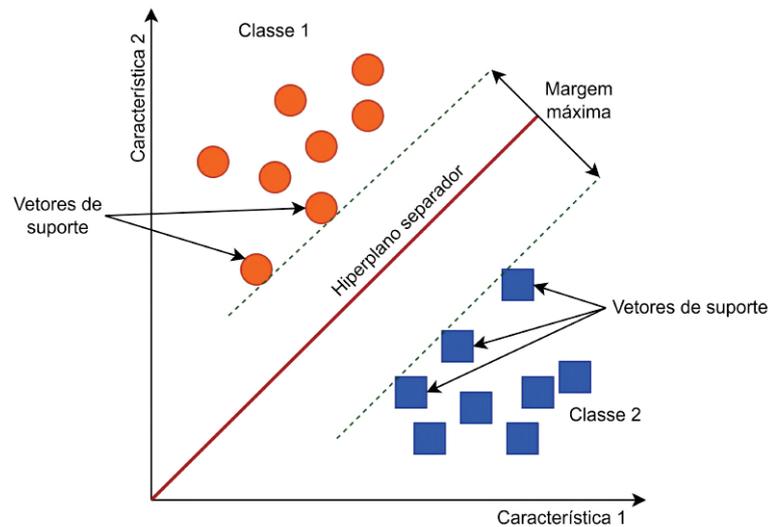
Exemplos de aplicação de regressão logística incluem estudos epidemiológicos (JIN et al., 2015), na classificação da eficiência de máquinas industriais (BORUCKA;

GRZELAK, 2019) e em assessoria financeira para tomada de decisão em relação a débitos imobiliários (STRZELECKA et al., 2020).

### 3.3.3 Máquina de vetor de suporte

Máquina de vetor de suporte é um tipo de classificador que busca separar os dados através de um hiperplano linear. A ideia de um modelo de SVM é ilustrada na FIGURA 10. Na FIGURA, é apresentado um problema de classificação binário separado por um hiperplano separador. Este hiperplano é definido através do cálculo de uma margem máxima de separação entre as classes. Esta margem máxima, por sua vez, é calculada pela seleção dos chamados vetores de suporte, isto é, dados diferentes de cada classe que maximizam esta margem (RUSSELL; NORVIG, 2021; GÉRON, 2019).

FIGURA 10 – ESTRUTURA DE UMA MÁQUINA DE VETOR DE SUPORTE



FONTE: Fonte: o autor

LEGENDA: Estrutura de máquina de vetor de suporte

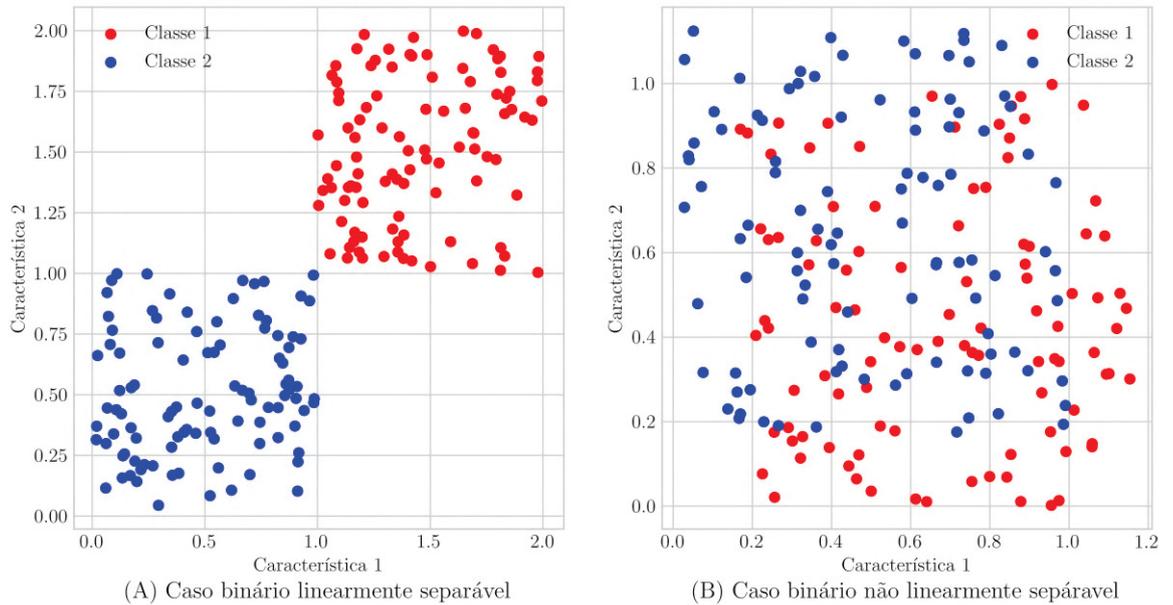
Na FIGURA 11 são ilustrados dois tipos problemas de classificação binário linearmente e não linearmente separável. Em (A) é o caso separável, enquanto em (B) o caso não separável.. Considerando o caso separável, para um conjunto de treinamento  $T$  em que uma amostra corresponde a  $(x_i, y_i)_{i=1}^N$ , em que  $N$  é o total de dados, em que os rótulos são representados como  $y_i = 1$  ou  $y_i = -1$ , a equação do hiperplano separador é dada por (RUSSELL; NORVIG, 2021):

$$w^T \mathbf{x} + b = 0 \quad (3.12)$$

Em que  $w^T$  é um vetor de pesos ajustável,  $\mathbf{x}$  é o vetor de entrada dos dados e  $b$  é um viés (*bias*, em inglês). Portanto (RUSSELL; NORVIG, 2021):

$$w^T \mathbf{x} + b \geq 0, \quad \text{para } y_i = 1 \quad (3.13)$$

FIGURA 11 – TIPOS DE PROBLEMAS BINÁRIOS



FONTE: Fonte: o autor

LEGENDA: Exemplo de problema binário, em (A) um caso linearmente separável e em (B) um caso não linearmente separável

$$w^T \mathbf{x} + b \leq 0, \quad \text{para } y_i = -1 \quad (3.14)$$

O conjunto de pontos  $(x_i, y_i)$  que satisfazem igualdade para a inequação 3.13 ou 3.14 são chamados de vetores de suporte. Como existem infinitos hiperplanos que podem ser escolhidos, o hiperplano separador ótimo é aquele que maximiza a margem de separação dos dados, ou seja maximiza a distância  $M$  indicada na FIGURA 12. Na FIGURA, o hiperplano separador é evidenciado, com a o tamanho da margem de separação indicada de  $\frac{2}{\|\mathbf{w}\|}$ , ou  $2M$  (RUSSELL; NORVIG, 2021; HASTIE et al., 2009b). Logo:

$$\max_{b, w, \|\mathbf{w}\|=1} M$$

com as restrições:  $y_i(\mathbf{w} \cdot x_i + b) \geq M, i = 1, \dots, N$

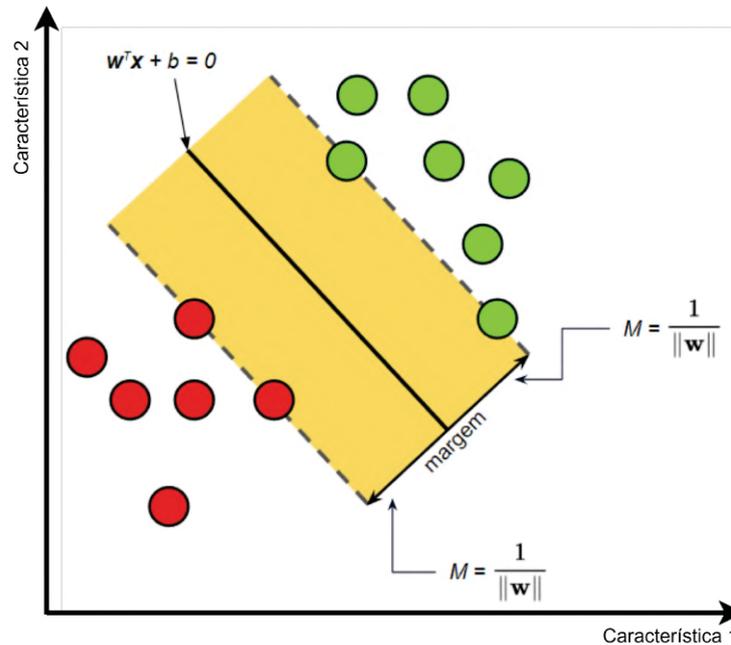
É possível reescrever o problema da seguinte forma (HASTIE et al., 2009b):

$$\min_{b, w} \mathbf{w} \quad (3.15)$$

$$\text{com as restrições: } y_i(\mathbf{w} \cdot x_i + b) + 1 \geq 0, i = 1, \dots, N \quad (3.16)$$

Que se trata de um problema de otimização quadrático. Existem diversas técnicas para resolução deste tipo de problema, como por exemplo o uso de multiplicadores

FIGURA 12 – DISTÂNCIA DA MARGEM DE SEPARAÇÃO DO HIPERPLANO DA SVM



FONTE: Fonte: Adaptada de (HASTIE et al., 2009b)  
 LEGENDA: Exemplo de hiperplano separador com a margem de separação

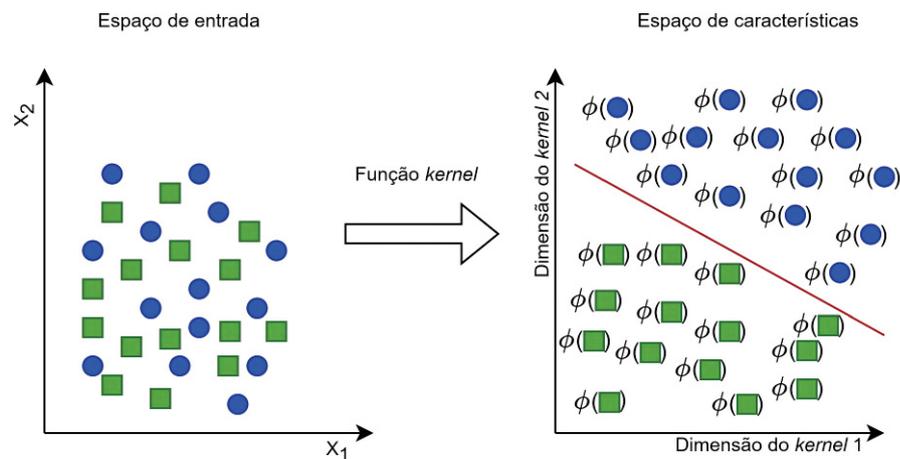
de Lagrange. A resolução leva a um classificador capaz de separar as duas classes (RUSSELL; NORVIG, 2021; HASTIE et al., 2009b).

Para um caso não separável, conforme ilustrado no exemplo (B) da FIGURA 11. Em problemas deste tipo, não é possível encontrar um hiperplano capaz de separar totalmente as duas classes. Para isto, é aplicado o chamado "truque do *kernel*", isto é, realizar um mapeamento dos dados  $\mathbf{x}$ , atualmente situados no chamado espaço de entrada, para um novo espaço de maior dimensão, denominado espaço de características. O "truque do *kernel*" consiste, então, em aplicar uma função *kernel*,  $\phi(\mathbf{x})$ , e transformar os dados em espaços de maior dimensão, possibilitando a construção de hiperplanos mais complexos que possibilitem separar as duas classes ou que diminuam o erro de classificação. A FIGURA 13 mostra um esquema simplificado deste processo. No exemplo da FIGURA, é aplicado o "truque do *kernel*" em um problema não linearmente separável, criando um cenário em que, num espaço de maior dimensão (simplificado em duas dimensões na FIGURA), os dados são, agora, separáveis (RUSSELL; NORVIG, 2021).

Existem diferentes funções *kernel* que podem ser utilizadas. Três possíveis escolhas para função *kernel* são (HASTIE et al., 2009b):

- *Kernel* polinomial, que assume a forma  $\phi(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d$ , em que  $d$  é um hiperparâmetro configurado. Particularmente quando  $d = 1$ , obtém-se o chamado

FIGURA 13 – EXEMPLO DE APLICAÇÃO DO TRUQUE DO KERNEL



FONTE: Fonte: o autor

LEGENDA: Exemplo de aplicação do truque do kernel em um problema não separável

*kernel* linear.

- *Kernel* função de base radial gaussiana (abreviada para RBF, do inglês *Radial Basis Function*, em que geralmente se oculta o termo gaussiano), que é definida como  $\phi(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$ , para todo  $\gamma > 0$  e  $\gamma$  sendo também um hiperparâmetro pré definido e maior que zero.
- *Kernel* sigmoidal, que é  $\phi(\mathbf{x}_i, \mathbf{x}_j) = \tanh(k_1 \mathbf{x}_i \cdot \mathbf{x}_j + k_2)$  para alguns, mas não todos,  $k_1 > 0$  e  $k_2 < 0$ , que também são pré configurados como hiperparâmetros.

Alguns exemplos de aplicação de SVM são para predição tipo de membrana de proteínas (CAI et al., 2004), em conjunto com ANN para detecção de intrusos (CHEN et al., 2005) e para busca de imagens em banco de dados (WONG; HSU, 2006).

### 3.3.4 Redes neurais artificiais

As ANN são inspiradas nas ligações nervosas do cérebro humano e no mecanismo de sinapse e transmissão dos pulsos eletroquímicos entre os neurônios. Os neurônios são conectados com outros neurônios através dos axônios e dendritos e a área de conexão é chamada de sinapse. Ao reagir a estímulos externos, sejam estes estímulos do ambiente externo ou um estímulos vindo de outro axônio, o neurônio pode transmitir um pulso eletroquímico liberando determinados tipos de neurotransmissores na região da sinapse. Estes neurotransmissores podem amplificar, atenuar ou mesmo eliminar o sinal. Quando o sinal é transmitido, é passado para o próximo neurônio, que pode continuar ou parar a transmissão. Este processo se repete até que o sinal chegue ao cérebro, podendo ativar ou não diferentes regiões cerebrais, cada uma responsável por um determinado tipo de processamento (ABIODUN et al., 2018).

Partindo desta inspiração, uma ANN cria uma rede de neurônios artificiais que frente a um estímulo, neste contexto chamado de entrada, pode ou não transmitir este estímulo para frente. Existem dois tipos de rede, a chamada propagação direta (ou *feedforward*, em inglês), em que o sinal trafega em uma única direção pela rede, e a chamada recorrente, onde existem conexões que retroalimentam o sinal na rede (WU; FENG, 2018).

Um tipo empregado de arquitetura de um neurônio artificial é o *perceptron*, proposto originalmente por Frank Rosenblatt em 1957. Um *perceptron* possui como entradas números (representadas como  $x_1, x_2, x_3, \dots, x_n$ ), sendo cada entrada associada a um peso (representados como  $w_1, w_2, w_3, \dots, w_n$ ). O equivalente ao núcleo do *perceptron* calcula a soma ponderada das entradas ( $z = w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_nx_n = \mathbf{x}^T \mathbf{w}$ ), podendo ter ou não um viés (*bias* em inglês) adicionado, representado como  $b$  (DELASHMIT; MANRY et al., 2005).

O *perceptron* aplica algum tipo de função de ativação, representada como  $\phi(z)$ , sobre o resultado calculado por este "núcleo". As funções de ativação têm como objetivo transformar a soma ponderada em um resultado, normalmente em valores que vão de 0 a 1, mas existem exceções para outros valores, de acordo com o problema. Alguns exemplos de funções de ativação utilizadas são (DELASHMIT; MANRY et al., 2005; ABIODUN et al., 2018; WU; FENG, 2018):

- Função degrau, que assume a forma (GÉRON, 2019):

$$\text{degrau}(z) = \begin{cases} 0 & \text{se } z < 0 \\ 1 & \text{se } z \geq 0 \end{cases} \quad (3.17)$$

- Função logística ou sigmoide (RUSSELL; NORVIG, 2021):

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (3.18)$$

- Função de unidade linear retificada (ReLU, do inglês *rectified linear unit*), definida como (RUSSELL; NORVIG, 2021):

$$\text{ReLU}(z) = \max(0, z) \quad (3.19)$$

- A função ReLU possui uma versão chamada de suavizador positivo ou *soft-plus*, em inglês, que introduz uma leve não-linearidade nos cálculos (RUSSELL; NORVIG, 2021):

$$\text{softplus}(z) = \log(1 + e^z) \quad (3.20)$$

- A função tangente hiperbólica, na forma de (RUSSELL; NORVIG, 2021):

$$\tanh(z) = \frac{e^{2z} - 1}{e^{2z} + 1} \quad (3.21)$$

Após o cálculo da função de ativação, obtém-se uma saída, que pode ser binária (no caso de problemas de classificação binário) ou linear (no caso de problemas de regressão). Desta forma, a saída do *perceptron* pode ser representado como (GÉRON, 2019):

$$h_{\mathbf{w}}(\mathbf{x}) = \phi(\mathbf{x}^T \mathbf{w} + b) \quad (3.22)$$

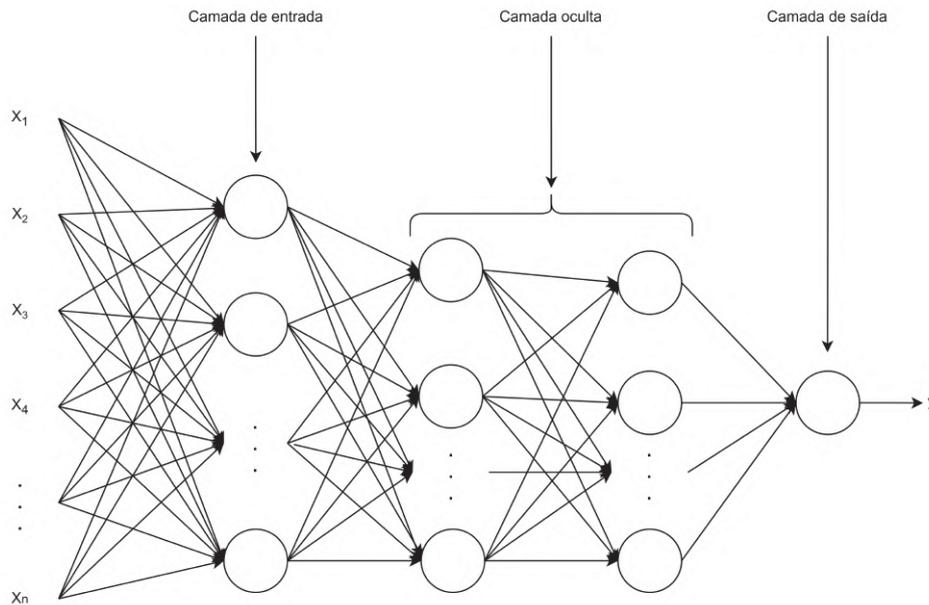
Com apenas um único *perceptron* a quantidade de saídas possíveis é limitada. Desta forma, em problemas de classificação, a fronteira de separação entre as classes possível de ser construída é apenas de um plano linear. No entanto, à medida que mais *perceptrons* são adicionados e novas camadas são construídas na rede, geometrias mais complexas são capazes de serem construídas com o treinamento adequado (WU; FENG, 2018).

Desta forma, quando se utiliza um modelo de ANN, são construídas uma ou mais camadas. Este tipo de modelo é chamado de multicamadas de *perceptrons* (ou *multi-layer perceptron*). A FIGURA 14 mostra um exemplo de arquitetura de rede multicamada, nesta FIGURA, os pesos e o viés (*bias*) foram ocultados para simplificar. Os *perceptrons* ligados aos dados de entrada compõe a chamada camada de entrada, com cada um dos neurônios ligados a todos os dados (ABDOLRASOL et al., 2021; PAUL et al., 2022).

Os últimos *perceptrons* compõe a camada de saída, no exemplo da FIGURA 14 esta camada é composta apenas por um neurônio, mas a quantidade de neurônios nesta camada pode variar de acordo com problema. Todos os neurônios entre a camada de entrada e a camada de saída, compõe a chamada camada oculta. Dentro da camada oculta, podem ser alocadas diferentes camadas, cada uma com uma quantidade diferente de neurônios. Quando esta camada é suficientemente grande, o modelo em questão passa a ser classificado com um modelo de DL. O conceito de suficientemente grande difere de autor para autor, sendo que alguns autores já consideram que mais do que duas camadas oculta compõe um modelo de DL, enquanto outros consideram que uma dúzia de camadas oculta ainda configuram uma rede rasa.

No exemplo da FIGURA, a camada oculta é composta de duas camadas, cada um com três neurônio. É importante ressaltar que a saída de uma camada é a entrada de outra camada e todos os neurônios estão conectados a todas as entradas, que é a forma mais usual de conexão para maioria das arquiteturas de rede (ABDOLRASOL et al., 2021; ABIODUN et al., 2018).

FIGURA 14 – MODELO DO TIPO MULTICAMADAS DE *PERCEPTRONS*



FONTE: Fonte: adaptada de (ABDOLRASOL et al., 2021)

LEGENDA: Arquitetura de um modelo do tipo multicamada de *perceptrons* com seis entradas, duas camadas na camada oculta e uma camada de saída.

O treinamento de ANN do tipo MLP é responsável por ajustar os valores dos pesos e do *bias*. Na primeira iteração do processo, os pesos e *bias* são inicializados com valores arbitrários. Então, cada um dos *perceptrons* é treinado com uma instância de dado por vez. Para cada neurônio que classifica de forma errada a saída, são reforçados as ligações e pesos que ajudaram a classificar de forma correta, desta forma, os *perceptrons* são treinados com base no erro de classificação da rede. A mostra equação 3.23 como o peso de um *perceptron* é atualizado em uma iteração do processo (GÉRON, 2019):

$$w_{i,j}^{\text{próxima iteração}} = w_{i,j} + \eta(y_j - \hat{y}_j)x_i \quad (3.23)$$

Na equação,  $w_{i,j}$  é a ponderação da conexão entre a entrada de número  $i$  do neurônio e da saída deste neurônio de número  $j$ .  $x_i$  é o valor da entrada de número  $i$  na iteração atual.  $\hat{y}_j$  é a saída de número  $j$  da iteração atual e  $y_j$  é o valor alvo da saída de número  $j$ , ou a saída dita correta.  $\eta$  é a taxa de aprendizado, um hiperparâmetro pré definido que tem como objetivo escalar o tamanho da atualização dos pesos, de modo

que as mudanças nos valores dos pesos sejam mais suaves, que facilita o treinamento e pode ajudar a evitar tanto o *overfitting* quanto *underfitting* (ABDOLRASOL et al., 2021; WU; FENG, 2018).

Este processo é feito por meio de uma técnica chamada de retropropagação (ou *backpropagation* em inglês), que consiste em aplicar o método do gradiente descendente em duas passagens pela rede. A primeira na propagação direta da rede, isto é da camada de entrada até a camada de saída e a segunda da camada de saída para a camada de entrada. De forma simplificada, em uma iteração do treinamento, a entrada é aplicada na rede de forma a se calcular o resultado. Com este resultado é calculado o erro, isto é, o quanto o resultado calculado diverge do valor real. Para calcular este erro, é previamente definido uma métrica, sendo normalmente empregada a função de perda de probabilidade logarítmica negativa (WU; FENG, 2018).

Com este erro, o gradiente descendente da rede é calculado, de modo que é possível medir a contribuição individual de cada peso e *bias* no erro. Este cálculo é realizado utilizando a regra da cadeia. Então, este erro é retropropagado para a rede, de modo a ajustar o valor dos pesos e *bias* e uma nova iteração do treinamento acontece. Todo o processo é repetido até que um número de iterações seja atingido (chamado de épocas) ou que a rede se estabilize (ABDOLRASOL et al., 2021; WU; FENG, 2018).

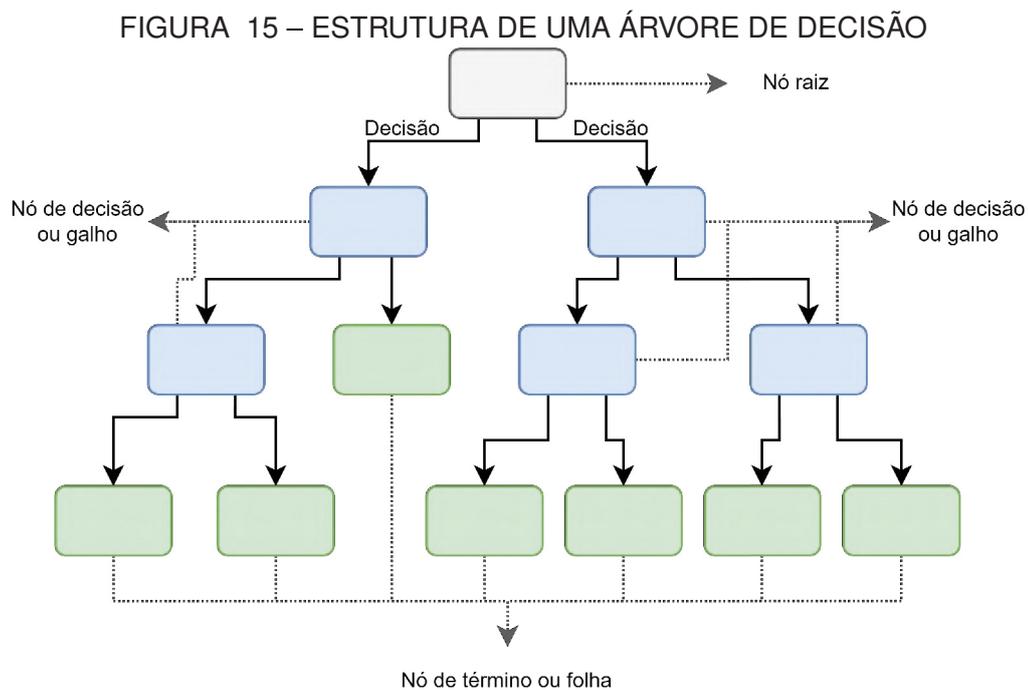
Ao final do treinamento, novas instâncias podem ser classificadas sendo alimentada pela rede no fluxo da camada de entrada para a camada de saída. Diferentes arquiteturas de rede podem ser construídas e existem outros tipos de neurônios e camadas, tal como recorrentes e convolucionais, para problemas mais complexos, tais como reconhecimento de imagens (GÉRON, 2019; RUSSELL; NORVIG, 2021). Se tratando de modelos MLP, alguns exemplos de aplicações são na previsão de preço e avaliação do mercado imobiliário (DANIELS; KAMP, 1999), na estimação da energia gerada por biomassa (PATTANAYAK et al., 2021) e na detecção de ataques em aplicações web (MOKBAL et al., 2019).

### 3.3.5 Modelos baseados em árvore

As próximas seções irão abordar os modelos baseados em árvores, como com as DTs, que constroem a base para outros modelos mais complexos como o AdaBoost, por exemplo. Alguns destes modelos também podem ser considerados modelos do tipo *ensemble*, mas como seu modelo base é um modelo do tipo árvores, estes modelos serão discutidos nestas seções, deixando as demais técnicas de *ensemble* utilizadas para a subseção 3.3.7, focada especificadamente nestes modelos.

### 3.3.5.1 Árvores de decisão

Os modelos de árvores são modelos versáteis, eficientes e de uma fácil interpretação e configuração, que podem ser utilizados tanto em problemas de classificação quanto em problemas de regressão. A ideia é construir um processo decisório hierarquizado em uma silhueta que remeta a uma forma de árvore. A cada passo, é avaliada uma expressão booleana até se chegar ao final da árvore. O processo se inicia em um nó raiz (ou *root node*), após a expressão booleana ser avaliada, o processo segue para os nós seguintes, chamados de nós de decisão ou galhos (em inglês *nodes* ou *branch*), em que novas expressões booleanas serão avaliadas. O processo continua até que se chegue a um nó de término ou folha (em inglês *leaf*), em que uma classe é atribuída e o processo é finalizado. A FIGURA 15 ilustra a esta estrutura básica de uma árvore de decisão, apontando o nó raiz, os nós de decisão e as folhas (HASTIE et al., 2009b; SOMVANSHI et al., 2016).



FONTE: Fonte: o autor

LEGENDA: Estrutura de uma árvore de decisão

As expressões booleanas são construídas durante o processo de treinamento para uma dada base de dados. A ideia básica é que seja escolhida uma característica específica da base de dados que produza um critério de separação que seja capaz de separar a maior quantidade de exemplos das classes possíveis. Uma forma de se medir isto é a partir da impureza. Dados ditos menos impuros são aqueles em que as classes pode ser separadas por aquele critério. Uma impureza igual a zero significa que aquele critério é perfeitamente capaz de separar as classes. Para isto, é escolhido

um método para se medir a impureza, tais como o índice de Gini, a entropia cruzada, ganho de informação, dentre outros (ALSAGHEER et al., 2017; MIENYE et al., 2019).

Para melhor exemplificar estes conceitos, seja uma base de dados que contenha  $n$  classes e uma classe em particular denotada como  $k$ . A 3.24 mostra como é calculado o índice de Gini

$$G_i = 1 - \sum_{k=1}^n p_{i,k}^2 \quad (3.24)$$

Nessa equação,  $G_i$  é o índice de Gini no nó de número  $i$  e  $p_{i,k}$  é a razão da classe  $k$  dentro da instância de treinamento no nó de número  $i$  (HASTIE et al., 2009b). Outra forma de calcular a impureza é a entropia cruzada,  $H_i$ , definida na EQUAÇÃO 3.25 (HASTIE et al., 2009b).

$$H_i = - \sum_{\substack{k=1 \\ p_{i,k} \neq 0}}^n p_{i,k} * \log(p_{i,k}) \quad (3.25)$$

Tanto o índice de Gini quanto a entropia cruzada produzem árvores com boas taxas de acerto, sendo o índice de Gini computacionalmente mais rápido, enquanto, para certos tipos de bases de dados, a entropia cruzada pode produzir árvores ligeiramente mais balanceadas (MIENYE et al., 2019). Outras formas de calcular a impureza também são presentes na literatura.

O processo de treinamento basicamente busca a expressão booleana com menor impureza e divide a base de treinamento em duas partes, conforme ilustrado na FIGURA 15, e continua esta busca para os nós adjacentes. O processo termina se é atingido uma profundidade máxima pré configurada ou quando o algoritmo não consegue encontrar nenhuma expressão capaz de diminuir a impureza daquele nó. Outros hiperparâmetros controlam este processo de divisão dentro do algoritmo (PRIYANKA; KUMAR, 2020).

Devido à forma hierarquizada das árvores de decisão, este tipo de modelo é considerado do tipo caixa branca, em que é fácil interpretar as decisões feitas pelo modelo. Isto faz com que estes modelos sejam utilizados para entender melhor as relações das características presente na base de dados Os modelos de árvore de decisão também servem de base para outras técnicas de aprendizado de máquina, como as florestas aleatórias (descritas a seguir) (ALSAGHEER et al., 2017). Exemplos de aplicação de árvores de decisão incluem desde de classificação de e-mails de *spams* (HASTIE et al., 2009b), compra e venda de ações (MIRÓ-JULIÀ et al., 2010), em auxílio

diagnóstico em câncer de mama (AL-SALIHY; IBRIKCI, 2017), reconhecimento de imagens (LU; YANG, 2009), entre outros.

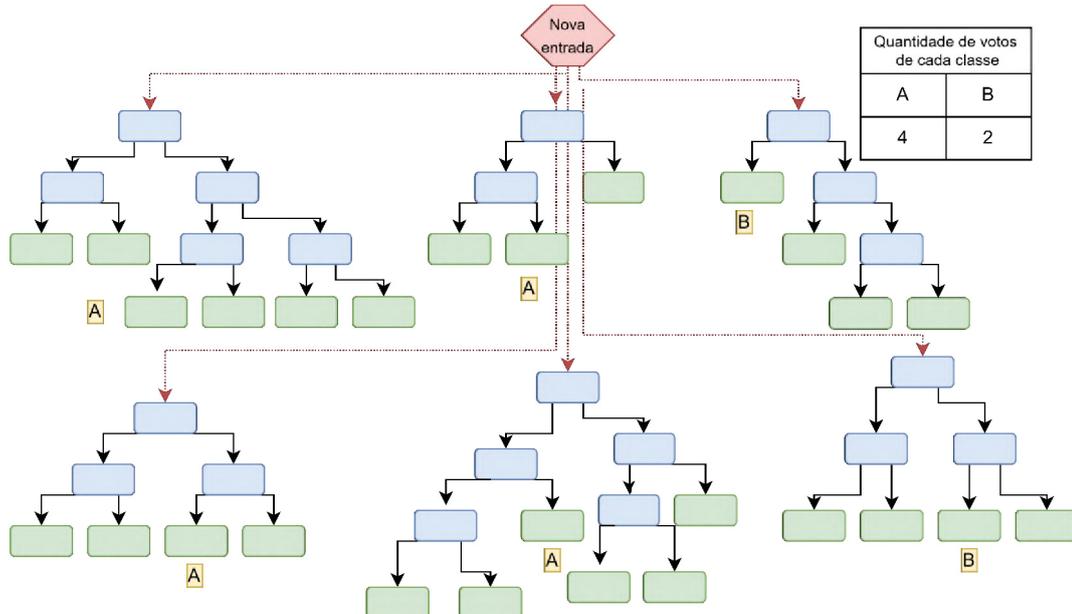
### 3.3.5.2 Floresta aleatória

Floresta aleatória (*random forest* em inglês) é um método de ML que envolve a montagem e agregação (*ensemble*) de diversas árvores de decisão, originalmente proposta por Breiman (2001). O método busca construir árvores de decisão decorrelacionadas utilizando a técnica de re-amostragem (*bootstrap*) (PARMAR et al., 2019)

O *bootstrap* consiste em gerar bases de dados construídas a partir de uma amostragem aleatória da base original. Diferentes árvores de decisão são criadas a partir destas novas bases seguindo um algoritmo similar ao descrito na subseção anterior (3.3.5.1). O modelo final é um aglomerado de árvores, em que novas entradas são avaliadas por todas as árvores e o resultado final é aferido a partir das observações de cada árvore (KULKARNI; SINHA, 2013).

A FIGURA 16 ilustra o modelo final e este processo para um problema de classificação das classes "A" e "B". Na FIGURA, os nós são representados pela cor azul, enquanto as folhas pela cor verde. A nova entrada, em vermelho, é direcionada a todas as árvores e o resultado de cada árvore, destacado em amarelo, é compilado na TABELA "Quantidade de votos de cada classe" da FIGURA. Para este exemplo, o rótulo final da nova entrada é "A", pois o rótulo que apareceu mais vezes nas árvores.

FIGURA 16 – ESTRUTURA DE UMA FLORESTA ALEATÓRIA



FONTE: Fonte: o autor

LEGENDA: Estrutura de uma floresta aleatória

A técnica de *bootstrap* gera bases menores utilizando uma amostragem aleatória com reposição, isto é, alguns exemplos podem aparecer mais de uma vez. Dessa

forma, alguns exemplos não estarão presente em nenhuma das bases, sendo estes exemplos separados para uma base denominada como *out-of-bag* (fora da amostra, em tradução livre). O algoritmo da floresta aleatória utiliza estas novas bases para construir as árvores de decisão introduzindo uma aleatoriedade no processo. A busca da característica para dividir um nó é feita por meio de um conjunto de características selecionados aleatoriamente da base de dados. A partir deste conjunto de características, o algoritmo irá buscar uma expressão booleana conforme descrito em 3.3.5.1 ou de forma aleatória. Este processo se repete até atingir o limite máximo de folhas e o total de árvores configurado. O algoritmo pode ainda utilizar a base *out-of-bag* como uma forma de validação durante o processo de treinamento, diminuindo o erro de classificação da floresta (FAWAGREH et al., 2014; HASTIE et al., 2009b).

RF é um método que aumenta a diversidade e diminui o viés das árvores de decisão. Também são utilizados em análises da importância de características (KULKARNI; SINHA, 2013). Alguns exemplos de aplicação de RF: em física para medidas fotométricas (CARLILES et al., 2010); na detecção de acidentes de trânsito (DOGRU; SUBASI, 2018) e em sistemas de medição de saúde (KAUR et al., 2019).

### 3.3.6 *Boosting*

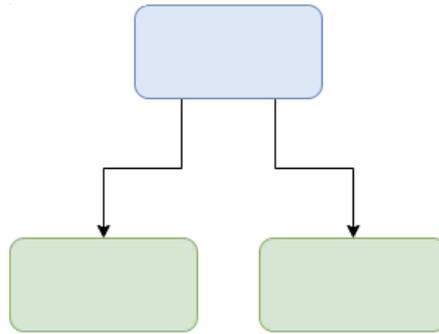
Os métodos de *boosting* foram originalmente criados para problemas de classificação, mas depois adaptados para problemas de regressão também. São métodos que combinam (fazem um *ensemble*) modelos (as vezes chamados de estimadores, em inglês *weak learners*) fracos em um comitê de decisão robusto. Os chamados estimadores fracos são definidos como modelos cuja taxa de acerto é apenas um pouco maior que classificações feitas de forma aleatória, um exemplo de modelo fraco é um "tronco"(em inglês *stump*), uma árvore de decisão que contenha apenas um nó, como mostrado na FIGURA 17. A proposta do *boosting* é criar uma sequência destes estimadores fracos, treinados com versões modificadas de dados. Uma nova entrada é, então, apresentada a todos estes estimadores, de modo que um comitê avalia o resultado final (HASTIE et al., 2009b; GÉRON, 2019).

Existem diferentes métodos de *boosting*, as próximas seções apresentam os utilizados neste projeto: *adaptive boosting*, *light gradient boosting* e *extreme gradient boosting*.

#### 3.3.6.1 *Adaptive boosting*

Originalmente proposto por Freund e Schapire (1995), o AdaBoost consiste em treinar uma série de classificadores fracos, partindo de um algoritmo base, como DT, ou SVMs. Para isto, o algoritmo atribui pesos as amostras da base de dados e busca construir um classificador simples. Então atribui um valor relativo ao quanto o

FIGURA 17 – ÁRVORE DE DECISÃO TIPO TRONCO



FONTE: Fonte: o autor  
 LEGENDA: Árvore de decisão tipo tronco

classificador foi capaz de separar as classes e utiliza o erro de predição deste classificador para alterar os pesos atribuídos. Em seguida, constrói um segundo classificador a partir dos pesos atualizados, calculando o valor relativo do poder de separação deste classificador e utilizando o erro de predição para atualizar os pesos das amostras. O processo continua até uma condição seja atingida, normalmente o total de estimadores a serem construídos (FERREIRA; FIGUEIREDO, 2012; SCHAPIRE, 2013).

Uma nova entrada é avaliada pelo AdaBoost sendo atribuída a esta sequência de classificadores. A cada iteração, é computado o resultado daquele classificador, sendo a classificação final definida pela soma do peso de cada classificador. A FIGURA 18 mostra este esquema para  $n$  classificadores.

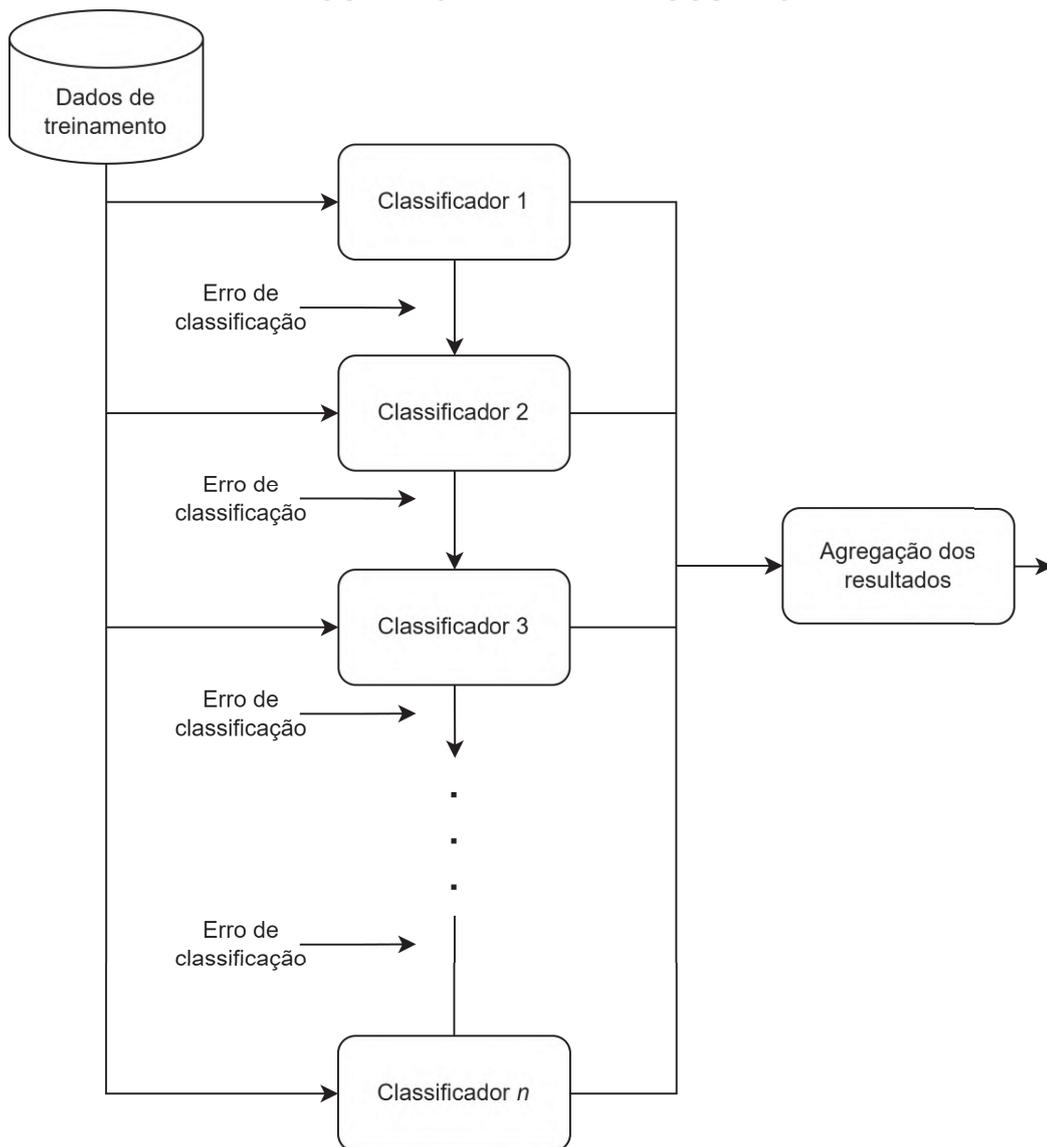
O AdaBoost foi utilizado em problemas de aprendizado *on-line*, como no trabalho de Fan et al. (1999). Em Zhu et al. (2023) foi aplicado na predição de carga elétrica utilizada em ambiente industrial. Akbar et al. (2020) aplicou o AdaBoost no auxílio diagnóstico e detecção de hepatite no trabalho.

### 3.3.6.2 *Gradient boosting*

O método do *gradient boosting* (GBO) e suas diferentes implementações possuem algumas similaridades com o método AdaBoost (descrito na subseção 3.3.6.1) e com a regressão logística (comentado na subseção 3.3.2). Assim como no Adaboost, diversas DT são construídas, mas em contrapartida, estas DT não são limitadas a apenas um nó e duas folhas (tronco ou *stump*), podendo ter tamanhos maiores. O GBO também busca construir DT com base no erro dos modelos anteriores, mas diferentemente do AdaBoost, ao invés de atribuir pesos aos rótulos da base de dados, o GBO cria DT a partir dos resíduos da classificação (NATEKIN; KNOLL, 2013).

Proposto em 1999 por Friedman (2001) e Friedman (2002), o algoritmo se inicia com uma única folha, que contém a probabilidade média dos rótulos da base de treinamento. Isto é feito com base na equação da regressão logística 3.9. A seguir,

FIGURA 18 – ADAPTIVE BOOSTING



FONTE: Fonte: o autor

LEGENDA: Estrutura do Adaptive Boosting

são calculados os resíduos da classificação dos dados de treino com base nesta única folha. Estes resíduos são, então, utilizados para construir uma DT, cujo o tamanho é configurado previamente. É importante notar que a DT construída visa classificar o *resíduo* e não os rótulos da base de treino. Os resíduos desta nova DT são calculados da seguinte forma:

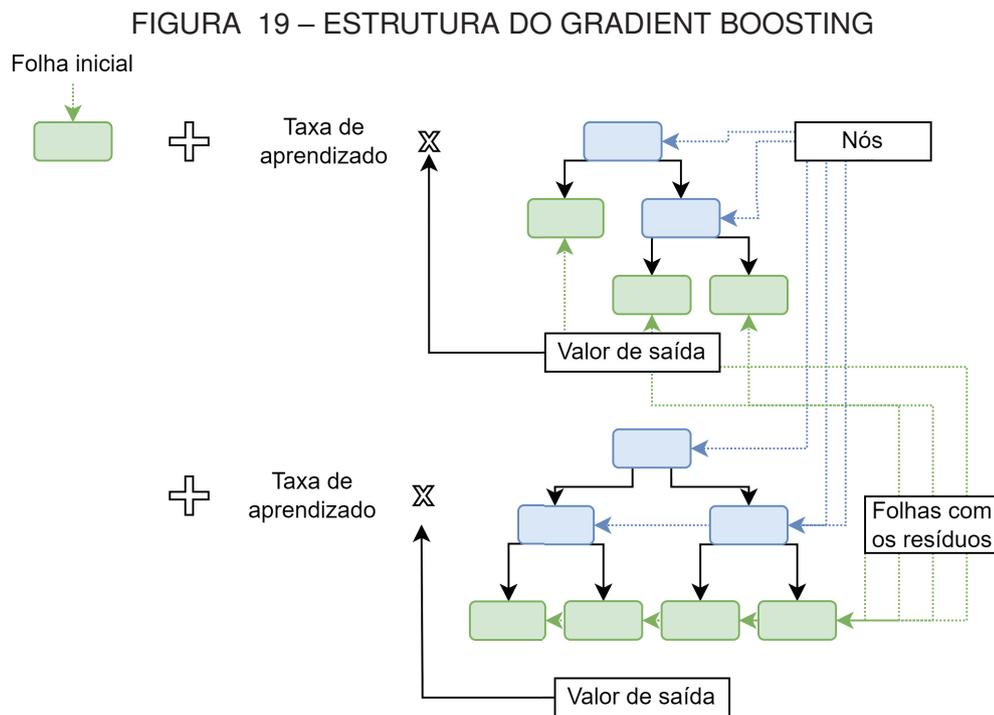
1. são calculados os valores de saída da DT, que convertem os resíduos das folhas em valores numéricos de acordo com a seguinte equação:

$$\text{Valor de saída} = \frac{\sum \text{Resíduo}_i}{\sum [\text{Prob. do nó anterior}_i \times (1 - \text{Prob. do nó anterior}_i)]} \quad (3.26)$$

2. Estes valores de saída são multiplicados pela taxa de aprendizado, que é configurada previamente e tem como objetivo diminuir a variância do modelo.

3. Por fim, estes valores são somados aos valores das demais DT e da folha inicial, sendo então a probabilidade final de classificação.

Então, o processo continua com o cálculo das diferenças da probabilidade original menos esta probabilidade final, chamados de resíduos, e uma nova DT é calculada a partir destes resíduos, seguindo os mesmos passos. A cada iteração do algoritmo, estes resíduos tendem a ficar menores e o algoritmo para quando não houver mudança nos resíduos ou atingir um limite de modelos ou interações previamente configurado. A FIGURA 19 exemplifica este processo para um total de duas DT. É destacado a folha inicial, os galhos e as folhas que contém resíduos. A FIGURA também ilustra o processo de cálculo da probabilidade, em que cada valor de saída de cada DT é multiplicado pela taxa de aprendizado e somado à probabilidade da folha inicial (GÉRON, 2019; HASTIE et al., 2009b).



Existem diferentes implantações do GBO, neste trabalho são utilizadas duas: o *extreme gradient boosting* (XGBoost) proposto por Chen e Guestrin em 2016 (CHEN; GUESTRIN, 2016) e o *light gradient boosting* (LightGBM) proposto por Ke et al (KE et al., 2017). O XGBoost utiliza algoritmos para melhor lidar com dados esparsos e uma estratégia de resumo de dados durante o aprendizado de árvores, chamada de "esboço de quantil ponderado" (ou *weighted quantile sketch* no original em inglês), que busca condensar os dados que serão utilizados no treinamento do modelo. Além destas implementações, o XGBoost trás algumas otimizações computacionais para operações

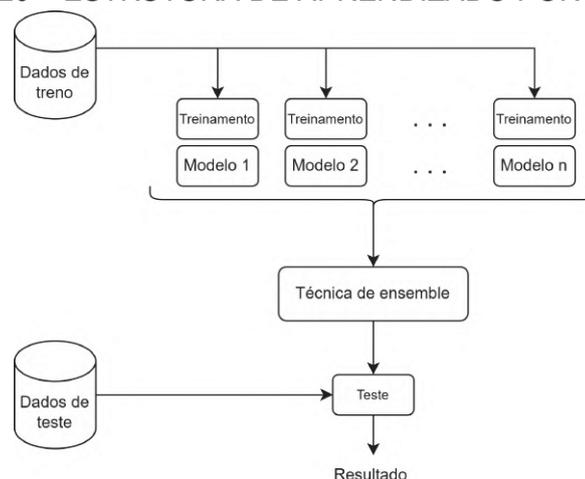
com grandes conjuntos de dados. O LightGBM, por sua vez, busca diminuir o tempo necessário para treinar um modelo do tipo do GBO, para isto esta implementação trás estratégias para calcular o ganho de informação com objetivo de diminuir a base de dados e para reduzir a quantidade de características (CHEN; GUESTRIN, 2016; KE et al., 2017).

Algumas das aplicações do XGBoost incluem análise de crédito Chang et al. (2018), predição de leucemia Ramaneswaran et al. (2021) e análise de sentimentos Afifah et al. (2021). Enquanto o LightGBM foi utilizado em problemas de avaliação de segurança alimentar Jing et al. (2022), análises farmacológicas Ota e Yamashita (2022), análise da qualidade da água Wagle et al. (2020), entre outros.

### 3.3.7 Modelos do tipo *ensemble*

Aprendizado por conjunto ou, *ensemble* em inglês, de modelos consiste em uma técnica que combina diferentes tipos de modelos para geração de um resultado. Algumas das técnicas já discutidas se enquadram como técnicas de *ensemble*, como a RF e as técnicas de *boosting* (descritos respectivamente nas subseções 3.3.5.2 e 3.3.6). Existem diferentes estratégias de combinação de modelos para criar um aprendizado por conjunto, que variam de acordo com o tipo de problema em questão (DONG et al., 2020). A imagem FIGURA 20 ilustra uma estrutura típica de *ensemble*. Na imagem, diferentes modelos de técnicas diferentes, treinados separadamente, combinam as previsões realizadas produzindo um novo resultado.

FIGURA 20 – ESTRUTURA DE APRENDIZADO POR CONJUNTO



FONTE: Fonte: o autor

LEGENDA: Estrutura do aprendizado por conjunto

Existem dois principais motivos para se utilizar alguma técnica de *ensemble*, reduzir o viés e a variância do resultado final. O tipo de modelo e o espaço de busca de hipóteses que o modelo impõe, pode acabar criando um viés de comportamento do

resultado, que pode ser diminuindo adicionando outros modelos para serem treinados dentro daquele problema. A variância também diminui, pois com mais modelos, para que se produza um resultado errado, é necessário que muitos (às vezes a maioria) dos modelos treinados produzam previsões incorretas para que o resultado final seja incorreto (assumindo que os modelos sejam independentes entre si), de forma que, com mais modelos, o resultado final é, de diferentes formas, ponderado pelos resultados individuais de cada modelo (RUSSELL; NORVIG, 2021). Nesta seção serão discutidas duas técnicas: voto majoritário e *bagging*.

### 3.3.7.1 Voto majoritário

Dentro de um problema de classificação, a técnica de VM consiste em combinar diferentes tipos de classificadores em um sistema de votação para produzir a previsão final. A ideia base nesta técnica é que, um resultado ruim de um classificador pode ser compensado com um resultado bom de outro classificador. Dois tipos de sistemas de votação são mais utilizados: majoritário (ou *hard voting* em inglês) ou ponderado por probabilidades (ou *soft voting*, em inglês). No sistema de voto majoritário, a previsão final é o rótulo mais atribuídos pelos classificadores (maioria dos votos), sendo possível atribuir pesos a cada classificador através de um hiperparâmetro do VM. Em caso de empate, o rótulo é atribuído seguindo uma lista decrescente dos classificadores (MOHANDES et al., 2018; PEDREGOSA et al., 2011). A FIGURA 21 exemplifica como o modelo funciona no caso de um não empate. Na FIGURA 21, a maioria dos classificadores atribuíram o rótulo de classe 3, portanto este é o resultado final do modelo VM.

No sistema ponderado por probabilidade, é calculada uma média ponderada da probabilidade de cada rótulo atribuído por cada classificador, sendo atribuído o rótulo de maior média. Um exemplo deste sistema é apresentado na TABELA 3. No exemplo da TABELA, após o cálculo das médias ponderadas da probabilidade (na TABELA,  $w_1$ ,  $w_2$ ,  $w_3$ ,  $w_4$  e  $w_5$  representam os pesos de cada classificador), a classe 2 obteve a maior média, sendo atribuída ao resultado final (MOHANDES et al., 2018; PEDREGOSA et al., 2011).

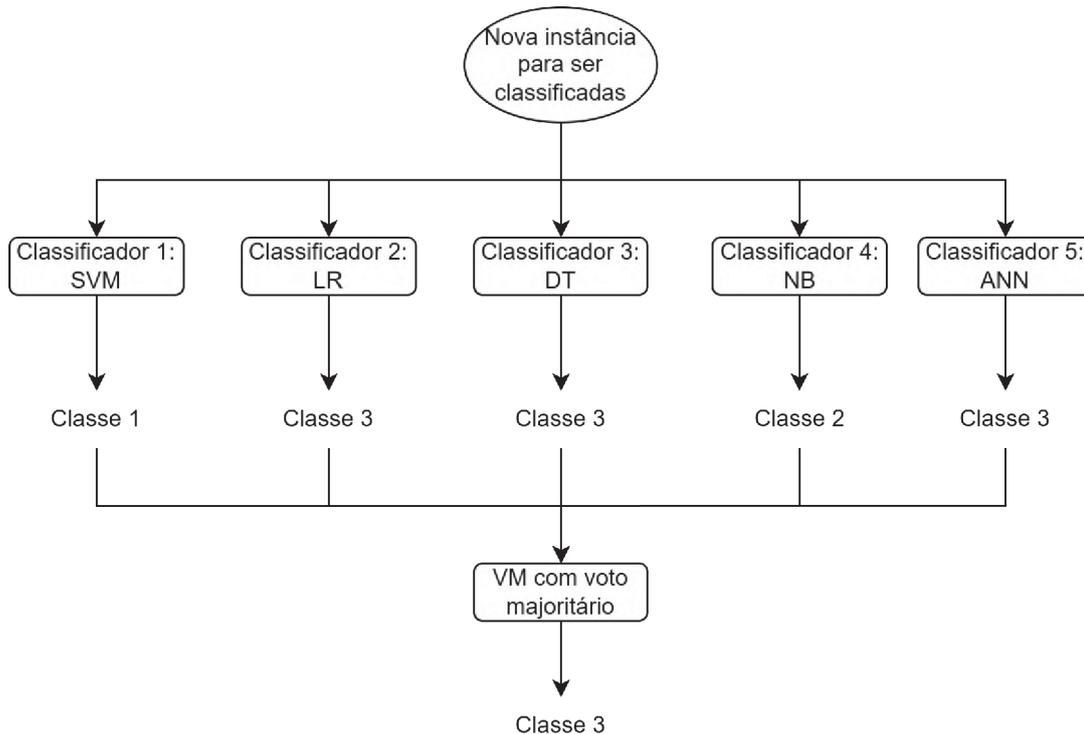
TABELA 3 – EXEMPLO DE VOTO MAJORITÁRIO COM SISTEMA PONDERADO

Classificador	Probabilidade classe 1	Probabilidade classe 2	Probabilidade classe 3
Classificador 1: SVM	$w_1 \times 0,20$	$w_1 \times 0,20$	$w_1 \times 0,60$
Classificador 2: LR	$w_2 \times 0,60$	$w_2 \times 0,30$	$w_2 \times 0,10$
Classificador 3: DT	$w_3 \times 0,30$	$w_3 \times 0,40$	$w_3 \times 0,30$
Classificador 4: NB	$w_4 \times 0,20$	$w_4 \times 0,70$	$w_4 \times 0,10$
Classificador 5: ANN	$w_5 \times 0,40$	$w_5 \times 0,55$	$w_5 \times 0,05$
Total ponderado	0,37	<b>0,45</b>	0,18

FONTE: Elaborada pelo autor

LEGENDA: Neste exemplo, a classe rotulada é a 2, pois possui o maior total ponderado.

FIGURA 21 – EXEMPLO DE VOTO MAJORITÁRIO COM SISTEMA MAJORITÁRIO



FONTE: Fonte: adaptada de (MOHANDÉS et al., 2018)

LEGENDA: Exemplo de voto majoritário com sistema majoritário

VM foi aplicado em diferentes problemas. Alguns exemplos são, classificação de cardiopatias Atallah e Al-Mousa (2019), predição de diabetes melitus Mushtaq et al. (2022) e predição de câncer de mama Naji et al. (2021).

### 3.3.7.2 *Bagging*

O classificador *bagging*, ou ensacamento em uma tradução livre, consiste em gerar  $K$  distintos subconjuntos de treinamento, construídos aleatoriamente e com repetição do conjunto de treinamento. Então, a partir de outro algoritmo base (como SVM ou DT, por exemplo), são treinados  $K$  modelos diferentes. Em um problema de classificação, a predição final segue a estratégia de voto majoritário descrito na técnica VM (3.3.7.1). O uso da técnica de *bagging* tende a reduzir a variância e consiste em uma abordagem útil quando a quantidade de dados é limitada ou quando o processo de treinamento do modelo base resulta em um modelo sobreajustado (*overfitted*) (RUSSELL; NORVIG, 2021).

Alguns exemplos de aplicação do classificador *bagging* são na detecção de intrusos Syarif et al. (2012), na predição de fraudes em cartão de crédito Zareapoor, Shamsolmoali et al. (2015) e em sistemas de hierarquização para redução de dimensionalidade em base de dados Datta et al. (2010).

A próxima seção aborda os tópicos referentes a diminuição de dimensiona-

lidade do espaço de características. Este tipo de técnica é utilizada com diferentes propósitos em um problema de modelagem com ML e pode ajudar a melhorar o desempenho dos modelos e tornar o tempo de treinamento menor das técnicas de modelagem.

### 3.4 REDUÇÃO DE DIMENSIONALIDADE

No contexto de ML, dimensionalidade é simplesmente a quantidade de características disponíveis para serem utilizadas no treinamento de um modelo em um problema específico. A redução de dimensionalidade consiste em um processo de combinar ou remover diferentes características, de forma a obter um conjunto de dados com menos características, mas tentando manter a mesma quantidade de informação relevante do problema (GÉRON, 2019).

Esta redução pode ser útil por diversos fatores, como por exemplo: facilitar a compreensão dos dados, ajudar na visualização dos dados e do problema, aumentar o tempo e a quantidade de espaço computacional para cálculo e armazenamento. Também pode ajudar a melhorar a acurácia dos modelos ao retirar informação irrelevante e diminuir o efeito da chamada "maldição da dimensionalidade", isto é, a dificuldade de alguns modelos de extraírem padrões em um conjunto de alta dimensão (GÉRON, 2019).

Nas próximas subseções são descritas técnicas de redução de dimensionalidade que combinam duas ou mais características em uma nova característica.

#### 3.4.1 Análise das Componentes Principais

A análise de componentes principais, PCA, é o método mais utilizado para redução de dimensionalidade e consiste em combinar as características nas chamadas componentes principais. O PCA é uma forma de aprendizado não-supervisionado que utiliza unicamente os dados de entrada, sem utilizar nenhuma referência dos rótulos do problema (caso existam). Ao invés disso, o critério é maximizar a verosimilhança (SALIH HASAN; ABDULAZEEZ, 2021).

Alguns conceitos chaves no PCA são a variância e covariância. Ambas são medidas do quão esparsos os dados estão em uma determinada base. Enquanto a variância é uma medida em uma única dimensão, a covariância mede a dispersão em duas dimensões, de forma que é possível avaliar se existe uma relação entre elas. A covariância de um par de duas características ( $X, Z$ ) que possuem valores  $(x_i, z_i)$  é calculada da seguinte forma mostrada na equação EQUAÇÃO 3.27. Para múltiplas

dimensões, é calculada a matriz de covariância (HASTIE et al., 2009b):

$$Cov(X, Y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(z_i - \bar{z})}{(n - 1)} \quad (3.27)$$

na equação 3.27  $\bar{x}$  é a média dos valores da características  $x$ , de modo análogo,  $\bar{z}$  é a média dos valores da característica  $z$  e  $N$  é o total de amostras.

O PCA é uma técnica de transformação linear que consiste em fazer uma projeção dos dados em um novo hiperplano. Este hiperplano é escolhido de forma que os eixos preservem o máximo de variância possível. O método do PCA calcula  $n$  componentes principais, em que  $n$  é a quantidade de características na base de dados. A chamada primeira componente principal é o eixo com maior variância dos dados, a segunda componente principal contém a segunda maior variância dos dados e assim por diante. Portanto a primeira componente principal é a que mais explica a variância dos dados e a ordem vai diminuindo de forma decrescente com as demais componentes (THARWAT, 2016).

O PCA é calculado utilizando a decomposição por valores singulares (DVS), que decompõe a matriz de treinamento  $\mathbf{X}$  em três diferentes matrizes de acordo com a equação EQUAÇÃO 3.4.1 (SALIH HASAN; ABDULAZEEZ, 2021; THARWAT, 2016; HASTIE et al., 2009b).

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T \quad (3.28)$$

Na equação a matriz  $\mathbf{U}$  contém os autovalores da matriz  $\mathbf{X}$ , a matriz  $\mathbf{D}$  contém os chamados valores singulares de  $\mathbf{X}$ . A partir desta matriz, é possível determinar o ranque da matriz  $\mathbf{X}$ , que corresponde ao total de diagonais não-zero da matriz.  $\mathbf{V}^T$  contém os vetores unitários que determinam as componentes principais (SALIH HASAN; ABDULAZEEZ, 2021).

Em um problema de modelagem, em que a base de dados é separada em treino e teste ou em treino, validação e teste, é importante que o PCA seja treinado apenas com a base de treino. Após o treinamento, é possível calcular o quanto cada componente contribui para a variância total dos dados, chamada de razão de variância explicada. Com base nisso, é possível determinar um corte do total de variância explicada necessária para o problema, mantendo as componentes principais que somem até este corte e eliminando as demais. O PCA é uma técnica paramétrica, que significa que uma vez calculadas as componentes principais da base de treino, é possível aplicar o mesmo cálculo para as bases de validação e teste. (THARWAT, 2016).

Para facilitar o entendimento, considere o exemplo ilustrado na TABELA 4. Neste exemplo, o problema contém 8 componentes principais e o valor da razão de

variância explicada de cada componente e a soma acumulada da razão de variância explicada total. Suponha que seja definido um corte de 0,95 de razão de variância total explicada. Neste caso, seria mantido até a componente 5 e as demais seriam descartadas. Assim, haveria uma redução de dimensionalidade de 8 dimensões para 5, ou 3 dimensões eliminada, mantendo 96% da variância dos dados originais.

TABELA 4 – EXEMPLO DE REDUÇÃO DE REDUÇÃO DE DIMENSIONALIDADE COM PCA

Componente	Variância explicada	Soma acumulativa da variância explicada
1	0,42	0,42
2	0,21	0,63
3	0,17	0,80
4	0,12	0,92
<b>5</b>	<b>0,04</b>	<b>0,96</b>
6	0,02	0,98
7	0,01	0,99
8	0,01	1,00

FONTE: Elaborada pelo autor

LEGENDA: Exemplo de redução de dimensionalidade com PCA para 8 componentes. No exemplo, seriam utilizadas até a componente 5, diminuindo a dimensão de 8 para 5.

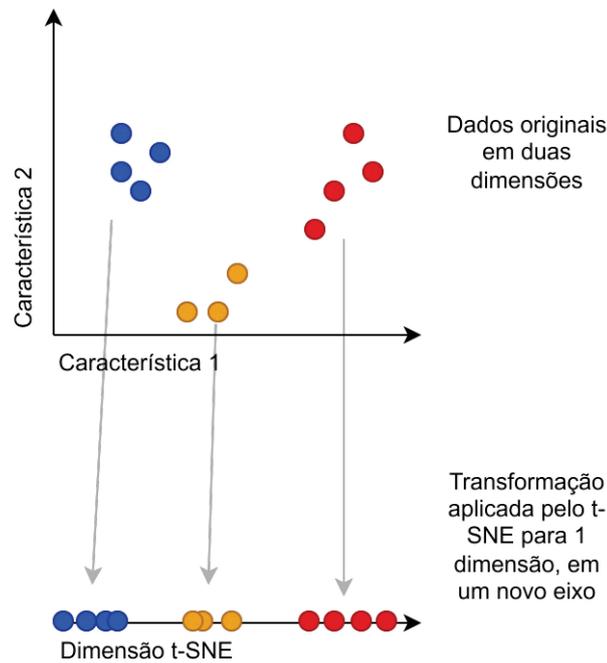
### 3.4.2 *t-Stochastic Neighbor Embedding*

O t-SNE é um método não-supervisionado que busca reduzir a dimensão dos dados de teste para uma, duas ou três dimensões tentando preservar a distância entre os dados e manter os dados vizinhos próximos proposto por Geoffrey Hinton et. al. nos trabalhos Van der Maaten e Hinton (2008) e Hinton e Roweis (2002). O t-SNE é amplamente utilizado na fase de exploração dos dados para facilitar a visualização de bases de dados de dimensões maiores que três. A FIGURA 22 ilustra um exemplo simplificado de aplicação do t-SNE. No exemplo, uma base de duas dimensões, com três grupos próximos, representados por cores diferentes, é transformada para uma dimensão, de forma que os dados vizinhos e a distância sejam preservadas (RUSSELL; NORVIG, 2021; VAN DER MAATEN; HINTON, 2008; HINTON; ROWEIS, 2002).

De uma forma simplificada, o algoritmo do t-SNE calcula uma matriz de similaridade entre os dados. Esta matriz é construída da seguinte forma (VAN DER MAATEN; HINTON, 2008):

1. Para um dado em específico é calculada a distância entre este dado e todos os demais.
2. Em seguida, as distâncias são projetadas em uma curva gaussiana centrada no dado específico.

FIGURA 22 – EXEMPLO DE UMA TRANSFORMAÇÃO APLICADA PELO MÉTODO T-SNE



FONTE: Fonte: o autor

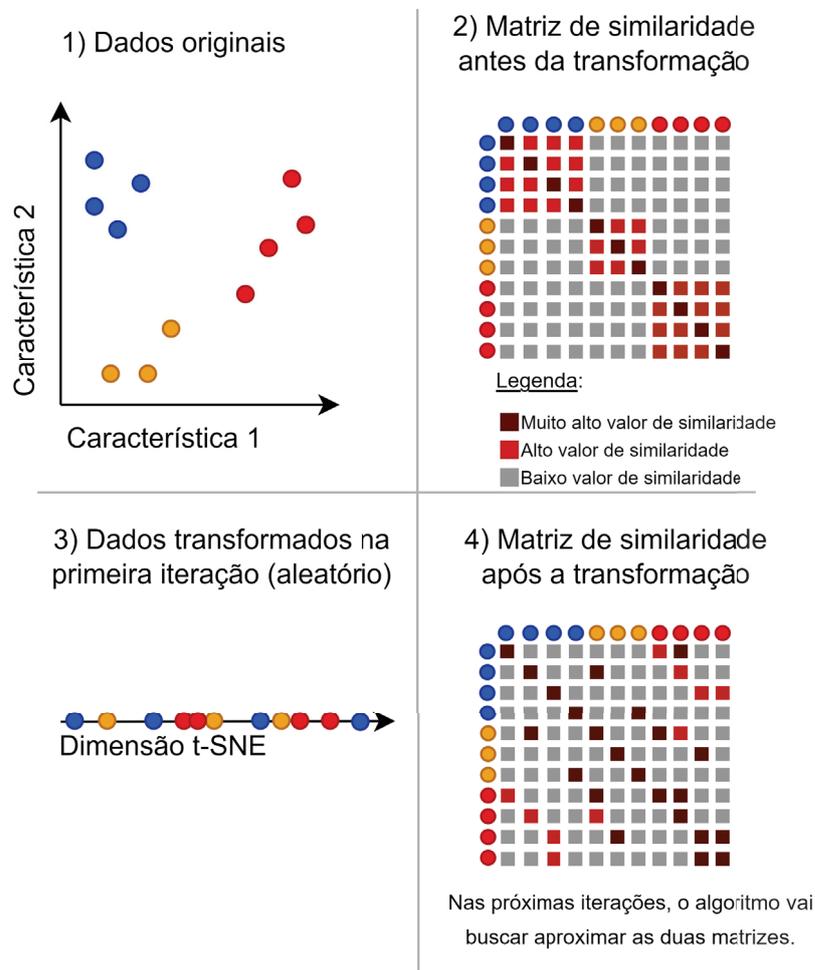
LEGENDA: Exemplo de transformação aplicada pelo método t-SNE

3. São medidas as distâncias destes pontos projetados com a curva gaussiana. Desta forma, pontos mais próximos do dado em questão possuirão maiores valores e pontos distantes possuirão menor valores, tendendo a valores muito baixos devido as características da curva gaussiana.
4. Estas novas medidas são escaladas para somarem 1, garantindo que curvas gaussianas de diferentes larguras terão similaridades comparáveis.
5. O processo é repetido para todos os dados.
6. Por fim, como a curva gaussiana é construída com base nos dados ao redor do dado específico, é possível que dois dados possuam valores de similaridade diferentes quando comparados bidirecionalmente. Neste caso, é calculado a média entre as similaridades.

Após o cálculo da matriz de similaridades, os dados são projetados no novo espaço aleatoriamente. Um novo processo de cálculo de similaridades acontece com os dados projetados, porém desta vez utilizando uma curva de distribuição t. Ao final do cálculo, é obtido uma nova matriz de similaridades com os dados transformados. O objetivo então é mover cada um dos pontos no novo espaço de modo que a nova matriz se assemelhe a matriz de similaridades original. O processo é repetido por uma quantidade pré definida de iterações. A FIGURA 23 mostra um exemplo da

primeira iteração deste processo. Na FIGURA, é calculado a matriz de similaridade dos dados originais com duas dimensões, desta forma cada um dos membros dos grupos ilustrativos possuem um alto valor de similaridade com os demais membros e baixo com os membros dos grupos opostos, razoavelmente distantes um dos outros. Em seguida, os dados são projetados aleatoriamente no novo espaço de uma dimensão e uma nova matriz de similaridade é calculada. Como os dados foram projetados aleatoriamente, a similaridade entre eles é diferente da calculada na primeira matriz. As próximas iterações do algoritmo, pequenos ajustes serão feitos na posição dos dados de modo que as matrizes se assemelhem (VAN DER MAATEN; HINTON, 2008).

FIGURA 23 – EXEMPLO DA PRIMEIRA ITERAÇÃO DO ALGORITMO T-SNE COM CÁLCULO DA MATRIZ DE SIMILARIDADE



FONTE: Fonte: o autor

LEGENDA: Exemplo da primeira iteração da transformação aplicada pelo método t-SNE com cálculo da matriz de similaridade

Diferentemente do PCA (descrito na subseção 3.4.1), o t-SNE originalmente é não paramétrico. Portanto não é possível aplicar a transformação para novas entradas, mas existem adaptações para tornar o método paramétrico (VAN DER MAATEN; HINTON, 2008; MAATEN, 2009).

### 3.4.3 *Uniform Manifold Approximation and Projection*

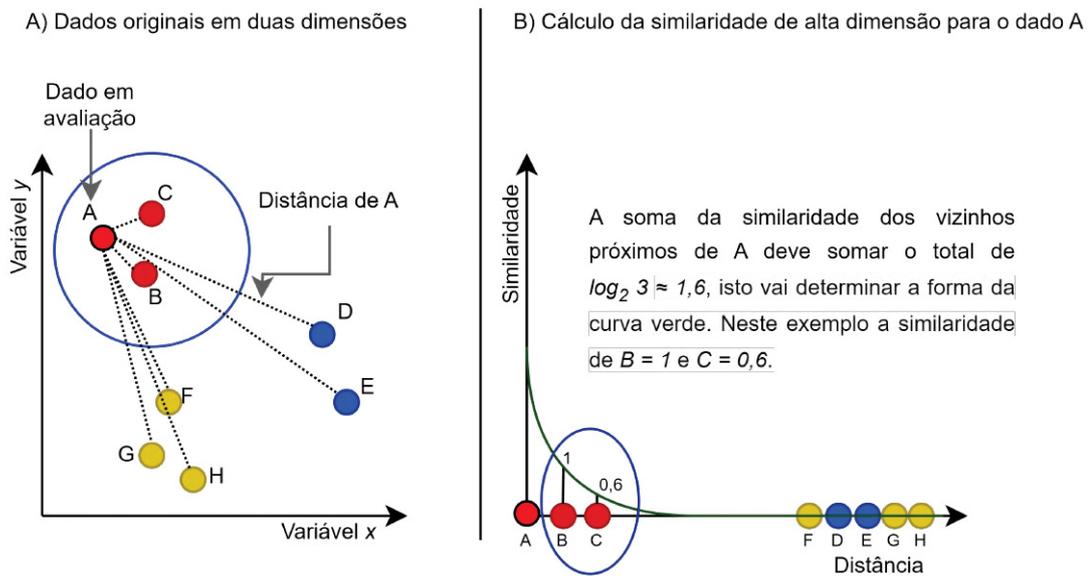
O método de redução de dimensionalidade UMAP foi proposto por McInnes et al. no artigo (MCINNES et al., 2018) e possui uma ideia semelhante ao t-SNE, descrito na subseção 3.4.2, que é reduzir uma base de dados de elevada dimensão para uma, duas ou três dimensões para facilitar a visualização dos dados. O objetivo do UMAP durante a transformação para uma dimensão menor é preservar a topologia dos dados (MCINNES et al., 2018).

A redução de dimensionalidade com o método UMAP se inicia com o cálculo da similaridade entre os dados. A similaridade para um dado específico é calculada a partir das distâncias entre este dado com todos os demais dados. Estas distâncias são projetadas em uma curva cuja forma depende de um hiperparâmetro pré-definido que é a quantidade de vizinhos desejada para cada dado. Para configuração da curva, os pontos do eixo  $y$  dos vizinhos mais próximos devem somar o logaritmo de base dois deste hiperparâmetro. Este valor é definido como a similaridade daqueles dados, chamado de score de alta dimensão. Este processo é repetido para todos os dados. Como para cada dado, a curva assumirá uma forma diferente, o valor da similaridade entre dois dados não é simétrico. A FIGURA 24 ilustra um exemplo de cálculo deste processo para um conjunto bidimensional de dados e número de vizinhos próximos igual a três. Na FIGURA, os cálculos de similaridades são feitos com base no dado rotulado como "A". Os demais dados são projetados na curva de acordo com as distâncias deles em relação à "A", portando os dados "C" e "B", que pertencem ao mesmo grupo, estão mais próximos de "A" que os pontos "D" e "H". Como o número de vizinhos próximos foi definido como 3, a soma da similaridade dos três vizinhos mais próximos de "A" (incluindo "A") deve totalizar um total de  $\log_2 3$ , que é aproximadamente igual a 1,6. Deste modo, a forma da curva será definida com o total das similaridades. Na FIGURA, esta é a curva verde indicada (MCINNES et al., 2018).

A próxima etapa do algoritmo é tornar os valores simétricos. Após finalizada os cálculos dos scores de alta dimensão, os dados são projetados no espaço de menor dimensão e ajustados de acordo com os scores de similaridade de alta dimensão. Para fazer este ajuste, dados que deveriam estar próximos (com alto score de alta dimensão) são escolhidos aleatoriamente e são movidos de modo a ficarem mais próximos e dados que deveriam estar longe também são escolhidos aleatoriamente e movidos para longe. Estes movimentos são feitos através do cálculo da similaridade neste novo espaço dimensional, chamada de score de baixa dimensão. A similaridade de baixa dimensão é calculada de forma análoga à de alta dimensão, com a diferença que as distâncias de baixa dimensão são projetadas em uma curva de distribuição t. O movimento então é guiado para maximizar o valor do score de baixa dimensão para os dados que estão próximos e minimizar o score de baixa dimensão para os dados

FIGURA 24 – EXEMPLO DE CÁLCULO DA SIMILARIDADE NO MÉTODO UMAP

Exemplo de cálculo de similaridade de alta dimensão para número de vizinhos próximos igual a 3



FONTE: Fonte: o autor

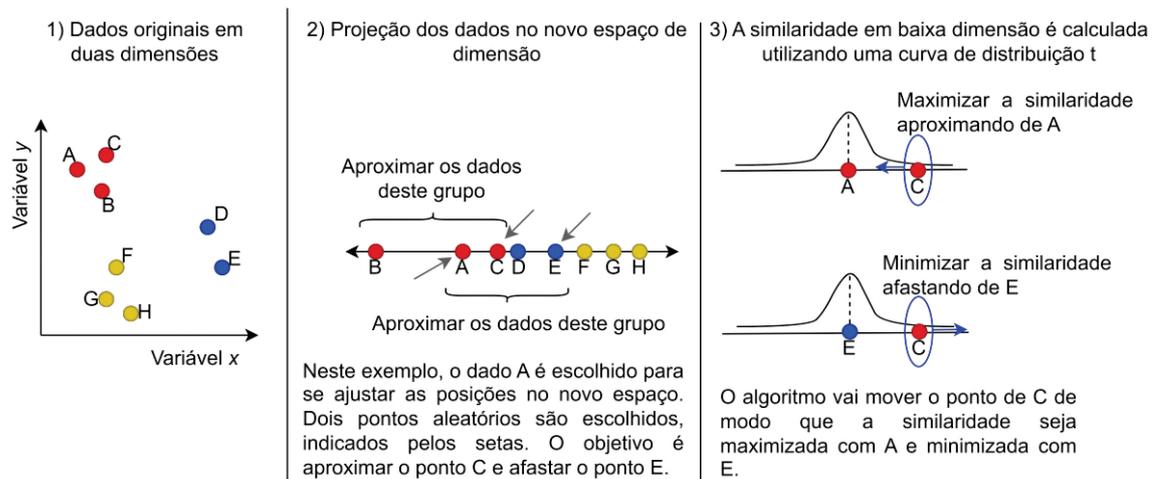
LEGENDA: Exemplo de cálculo de similaridade do método UMAP para números de vizinho próximos igual a 3

que estão distantes. Continuando o exemplo da FIGURA 24, a FIGURA 25 ilustra uma iteração deste processo. Primeiro, os dados originais de duas dimensões são projetados no espaço de dimensão menor, no caso uma dimensão. A segunda etapa é ajustar os pontos no novo espaço, isto é feito escolhendo um ponto aleatório, na FIGURA o ponto "A", e a partir deste escolhendo dois outros pontos para se moverem. Um ponto para se aproximar de "A" e um ponto para se afastar de "A". A escolha é feita com base nos grupos calculados com a similaridade do espaço de alta dimensão. Na FIGURA, estes os pontos são indicados pela seta, "C" e "E". A terceira etapa do processo é calcular a similaridade de "C" com "A", com objetivo de aproximar "C" de "A" e a similaridade de "E" com "C", para afastar "C" de "E". Diferentemente da forma calculada em alta dimensão, para baixa dimensão o cálculo é feito com uma curva de distribuição t. A curva é a mesma em todos os casos. Finalmente, o algoritmo procura maximizar a similaridade de "C" com "A" (aproximando os pontos) e minimizar a similaridade de "C" com "E" (afastando os pontos). A partir da segunda etapa, o processo se repete até que um número de iterações pré-definido, chamado de épocas, seja atingido (MCINNES et al., 2018).

Os métodos UMAP e t-SNE são semelhantes, mas possuem algumas diferenças cruciais. A primeira é que o método t-SNE inicializa (projeta) os dados no espaço de menor dimensão de forma aleatória, enquanto o UMAP utiliza uma técnica chamada de incorporação espectral (*spectral embedding* em inglês) para inicializar os dados

## FIGURA 25 – EXEMPLO DE AJUSTE DOS DADOS NO ESPAÇO TRANSFORMADO NO MÉTODO UMAP

Exemplo de ajuste de posição dos dados após transformação para redução de dimensão



FONTE: Fonte: o autor

LEGENDA: Exemplo de ajuste dos dados no espaço de menor dimensão com cálculo similaridade de baixa dimensão do método UMAP

no espaço de menor dimensão. Assim, enquanto diferentes execuções do t-SNE na mesma base de dados podem obter diferentes inicializações do espaço de menor dimensão, com UMAP é sempre obtido a mesma inicialização. Uma segunda diferença importante é que o t-SNE move todos os pontos ao ajustar sua localização no espaço de menor dimensão, enquanto o UMAP move apenas um ou um pequeno conjunto de pontos a cada iteração. Isso ajuda em termos de processamento computacional em bases com grande volumes de dados (MCINNEN et al., 2018; VAN DER MAATEN; HINTON, 2008).

A redução de dimensão é uma das formas de trabalhar com o problema de bases de dados com alta dimensão, tanto para visualização, quanto para processamento dos dados e melhora nos resultados dos modelos. As técnicas descritas nesta seção buscam manter a quantidade de informação relevante do problema enquanto condensam as características originais em um espaço de menor dimensão. A próxima seção aborda outro problema comumente enfrentado na modelagem por ML, a configuração e sintonia dos hiperparâmetros de cada modelo.

### 3.5 O PROBLEMA DA OTIMIZAÇÃO DOS HIPERPARÂMETROS

A busca do conjunto de hiperparâmetros para um dado modelo de ML é uma das tarefas que mais consomem tempo dentro das atividades de modelagem. É também uma das atividades de maior importância, uma vez que os hiperparâmetros desempenham um papel fundamental na acuracidade e capacidade de generalização dos modelos. Conjuntos não otimizados de hiperparâmetros podem levar à modelos

sobreajustados ou subajustados ao problema em questão (AKIBA et al., 2019; BISCHL et al., 2023; PROBST et al., 2019; VAN RIJN; HUTTER, 2018).

Os hiperparâmetros do modelo são diferentes dos parâmetros internos do modelo, que também passam por um processo de otimização. Enquanto os hiperparâmetros são configurados antes da etapa de treinamento e ajudam a direcionar o aprendizado do modelo ou mesmo estabelecer a arquitetura daquele modelo (como a quantidade de camadas de rede em uma ANN ou a quantidade de árvores em uma DT), os parâmetros interno ajustam as equações e pesos do modelo, que calculam a saída daquele modelo dada uma certa entrada (PROBST et al., 2019; VAN RIJN; HUTTER, 2018; WEERTS et al., 2020).

Durante a etapa de treinamento do modelo, o processo de aprendizagem do modelo em questão vai configurar os parâmetros internos do modelo de acordo com os dados de treinamento e teste. O método aplicado para este ajuste varia de modelo com modelo, podendo utilizar técnicas como gradiente descendente ou mesmo um algoritmo de otimização (PROBST et al., 2019; VAN RIJN; HUTTER, 2018).

Para otimizar e ajudar a configurar os hiperparâmetros, existem abordagens empíricas e de tentativa-e-erro, que podem depender muito do conhecimento do domínio específico do problema em questão. Estas abordagens podem levar a bons resultados quando bem empregadas, mas podem acabar restritos à base de dados utilizadas e ser necessário reanalisar os hiperparâmetros quando uma nova massa de dados está disponível. Também existem estudos que buscam criar um conjunto padrão de hiperparâmetros para serem utilizado em problemas diversos. Outra forma de configurar é através de softwares de métodos automatizados de aprendizado de máquina (AutoML, do inglês *automate machine learning*) e ferramentas de otimização como, por exemplo, *Hyperopt*, *Spearmint*, *SMAC*, *Autotune*, *Vizier*, *Optuna*, entre outros. Os métodos que utilizam ferramentas de softwares possibilitam que se determine um espaço de buscas, com uma maior amplitude de possibilidade de escolhas de hiperparâmetros. No entanto, o processo deve ser empregado com cautela, pois sem agregar conhecimento do domínio do problema, pode-se acabar novamente gerando modelos sobreajustados (PROBST et al., 2019; VAN RIJN; HUTTER, 2018).

Este trabalho utilizou a ferramenta Optuna. A próxima seção introduz, de forma simplificada, como o método opera.

### 3.5.1 Sintonia dos hiperparâmetros com Optuna

Em 2019, Akiba et. al. Akiba et al. (2019) propuseram uma nova ferramenta para otimizar os hiperparâmetros em modelos de ML. Os autores observaram que as ferramentas existentes possuíam limitações, como a necessidade de descrever detalhadamente o espaço de buscas, o que dificulta a execução de grandes experi-

mentos, e a falta de estratégias que possam diminuir o tempo de otimização, como o *pruning* (poda de soluções que já demonstraram não serem ótimas). Visando solucionar este problema, os autores elaboraram uma nova técnica chamada Optuna, buscando atender três princípios:

- Possibilidade de definir e alterar dinamicamente o espaço de busca durante a execução do programa de otimização,
- Algoritmo eficiente de amostragem e *pruning*, com possibilidades de customizações, de modo a diminuir o tempo de otimização,
- Facilidade de configurar e preparar a ferramenta, possibilitando tanto a execução em ambientes de um único computador ou sistemas distribuídos.

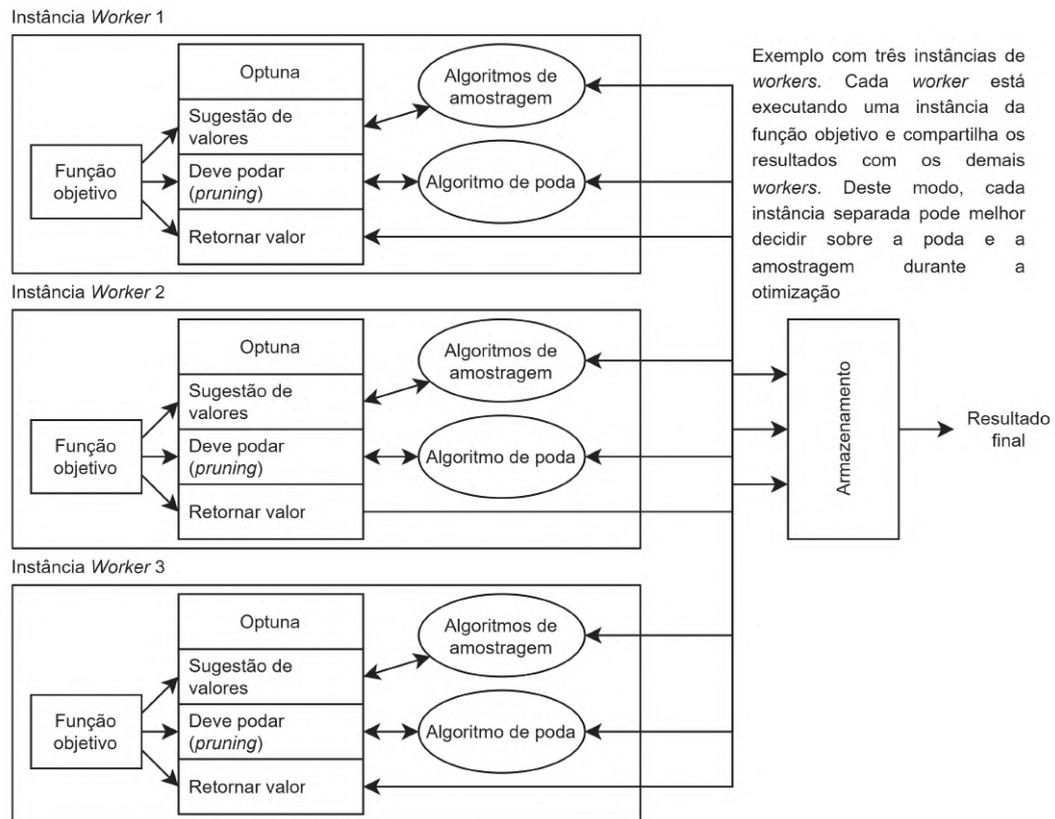
O Optuna formula um problema de otimização do tipo maximização/minimização com uma função objetivo que recebe um conjunto de hiperparâmetros como entrada e retorna um valor validação ou *score*. Tanto a função objetivo, quanto o conjunto de hiperparâmetros e o valor são definidos pelo usuário da ferramenta. O Optuna chama o processo de otimização de estudo (*study*) e cada iteração do processo de tentativa *trial* (AKIBA et al., 2019).

Durante o processo de otimização, o Optuna vai gradualmente construindo a função objetivo através das interações de cada *trial*. O espaço de buscas também construído dinamicamente ao longo do processo. A cada iteração, o *trial* faz uma *sugestão* dos hiperparâmetros dentro deste espaço de busca. Esta sugestão é baseada em dois métodos dentro do algoritmo: mecanismo de *pruning* e um método de amostragem. O sistema Optuna cria uma série de unidades de processamento chamada de trabalhadores (*workers*), cada *worker* executa uma instância da função objetivo independentemente e compartilha o progresso do *study* atual através de um sistema de armazenamento, este sistema é esquematizado na FIGURA 26 (AKIBA et al., 2019).

O mecanismo de *pruning* tem duas funções: 1) periodicamente monitorar os resultados intermediários da função objetivo e 2) finalizar um *trial* que não atende a uma condição pré-definida. O mecanismo é baseado no algoritmo *Asynchronous Successive Halving* (ou redução pela metade sucessiva assíncrona, em tradução livre) descrito em (JAMIESON; TALWALKAR, 2016; LI et al., 2018b). Este algoritmo possibilita que cada *worker* possa parar (podar) a execução de um *trial* baseado em uma classificação provisória de *trials*. Isto permite que nenhum *worker* precise esperar a finalização dos cálculos e resultados de outro *worker* (AKIBA et al., 2019).

O Optuna fornece diferentes métodos de amostragem para utilização. Dentre os métodos utilizados, quando um *study* possuir menos de mil iterações, é recomendado

FIGURA 26 – VISÃO GERAL DO SISTEMA OPTUNA



FONTE: Fonte: adaptada de (AKIBA et al., 2019)

LEGENDA: Visão geral do sistema Optuna

utilizar a técnica *tree-structured Parzen estimator* (TPE), ou estimador Parzen estruturado em árvore, em tradução livre, utilizado neste trabalho. O método TPE, descrito em (BERGSTRA et al., 2011), utiliza uma busca aleatória com dois métodos sequenciais gananciosos baseados maximizar o valor esperado, baseados em otimização bayesiana. Outros métodos de amostragem fornecidos pelo Optuna são CMA-ES (do inglês *covariance matrix adaptation-evolution strategy*, descrito em (HANSEN; OSTERMEIER, 2001)) e GP-BO (do inglês *gaussian process - bayesian optimization*, descrito em (SHAHRIARI et al., 2015)) (AKIBA et al., 2019; BERGSTRA et al., 2011).

O Optuna tem sido utilizado em diferentes tipos de trabalhos. Em 2023, Han e Oh Han e Oh (2023) utilizaram o Optuna dentro de um problema de AOS. Também dentro de aplicações em saúde, em 2022 Srinivas e Katarya aplicaram Optuna Srinivas e Katarya (2022) na predição de doenças cardiovasculares. Outras áreas do conhecimento também têm utilizado, como em 2020 que Domashova e Mikhailina Domashova e Mikhailina (2021) empregaram no contexto de detecção de lavagem de dinheiro.

A próxima seção aborda outro problema na modelagem por ML, a explicabilidade dos modelos e resultados. Mesmo com todo cuidado no processo de modelagem para garantir resultados estatisticamente coerentes, muitas vezes é necessário enten-

der porque tais decisões foram tomadas. Isto ajuda não somente a validar os resultados do ponto de vista conceitual do modelo, como também a aumentar o entendimento e conhecimento a cerca do problema.

### 3.6 EXPLICABILIDADE DOS MODELOS

Um dos grandes problemas em abordagens utilizando ML e DL é a explicabilidade do modelo construído. Quando é o modelo é relativamente simples, como por exemplo uma LR, é particularmente fácil computar e entender quais características mais impactaram no processo decisório do modelo. No entanto, quando o modelo é mais complexo, como uma árvore de grande profundidade ou uma rede de muitas camadas, esta tarefa pode se tornar complicada até para especialistas da área. Com o advento de tecnologias *big data* e a crescente capacidade em produzir e armazenar dados, modelos complexos têm sido mais utilizados, trazendo para discussão o equilíbrio entre acuracidade e interpretabilidade (LUNDBERG; LEE, 2017).

Trabalhando neste problema, em 2017 Lundberg e Lee Lundberg e Lee (2017) propuseram um método de análise da explicabilidade chamado SHAP, *SHapley Additive exPlanations*, ou explicações aditivas de valores *shapley*, em tradução livre. Como sugere o nome, o método se baseia nos valores *shapley* da teoria dos jogos.

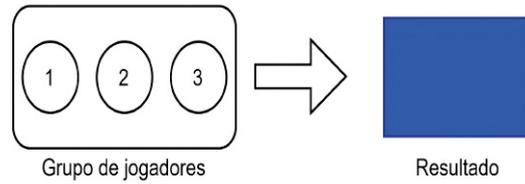
Os valores *shapley* são utilizados para explicar a contribuição individual de cada jogador (ou contribuinte) a um resultado obtido por um dado grupo de jogadores (ao qual, este jogador em questão faz parte). Para isto, é calculada a contribuição marginal de cada combinação em que um jogador específico está incluso ou não. O valor *shapley* é então calculado como a média destas contribuições marginais (HART, 1989).

Para melhor visualizar o conceito, considere este exemplo simplificado: em um grupo de três jogadores, denominados 1, 2 e 3, produziram um resultado de valor  $v$ . Para medir a contribuição individual do indivíduo 1, por exemplo, é utilizado o valor *shapley*. Durante o cálculo, é computado todas os valores  $v$  produzidos pelas contribuições marginais do jogador 1. Primeiro é calculado a contribuição de todos os jogadores e o valor produzido com os jogadores 2 e 3. A diferença entre o valor produzido pelo grupo todo e com ausência do jogador 1 é a contribuição marginal do jogador 1 na cooperação de 2 e 3. Este passo é ilustrado esquematicamente na FIGURA 27.

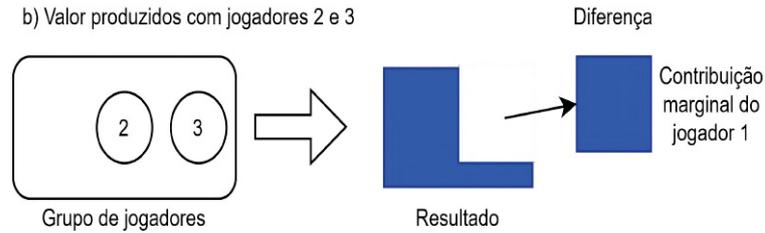
Este passo é repetido para todas as combinações contendo ou não o jogador 1, como mostra a FIGURA 28. A média das diferenças é o valor *shapley* do jogador 1, como mostra a FIGURA 29. O processo é então repetido para os jogadores 2 e 3, de modo que se calcule os valores *shapley* para todos os jogadores.

FIGURA 27 – CONTRIBUIÇÃO MARGINAL DO VALOR SHAPLEY

a) Valor produzidos com jogadores 1, 2 e 3



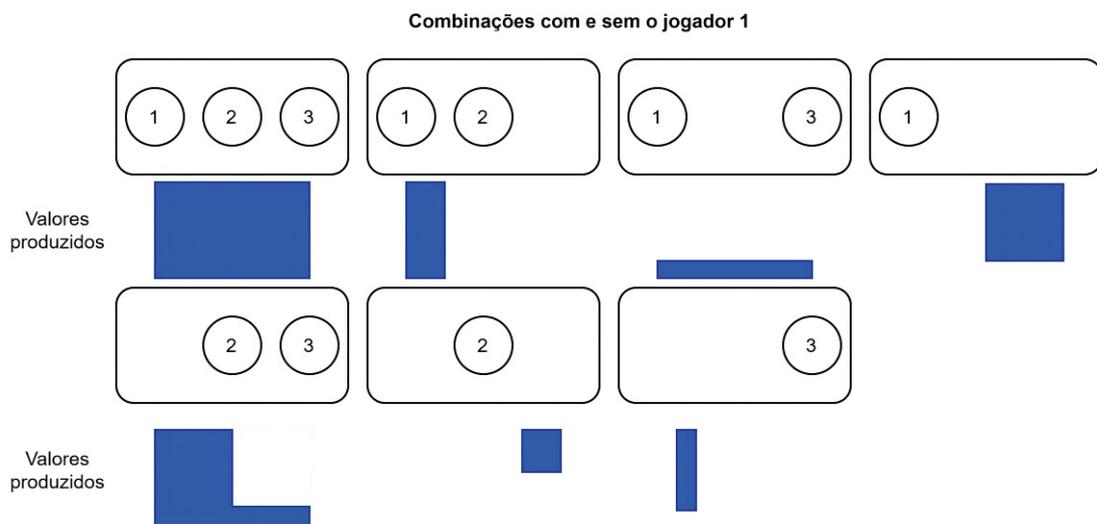
b) Valor produzidos com jogadores 2 e 3



FONTE: Fonte: o autor

LEGENDA: Contribuição marginal do jogador 1 na cooperação de 2 e 3

FIGURA 28 – CONTRIBUIÇÃO MARGINAL DO VALOR SHAPLEY DAS COMBINAÇÕES COM E SEM JOGADOR 1



FONTE: Fonte: o autor

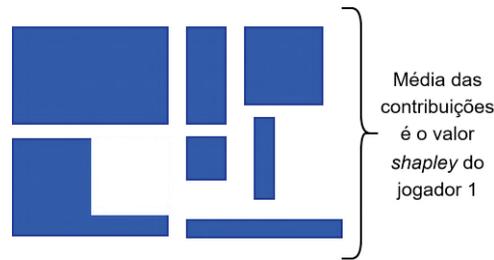
LEGENDA: Combinações de cooperações com e sem o jogador 1

De forma simplificada, o processo é descrito como (LUNDBERG; LEE, 2017; HART, 1989):

$$\phi_i = \frac{(1)}{\text{Total de jogadores}} \sum_{\forall C \text{ s.t. } i \notin C} \frac{\text{Contribuição marginal de } i \text{ para } C}{\text{Total de cooperações de tamanho } C} \quad (3.29)$$

Em que  $C$  representa uma combinação de cooperação,  $\phi$  é o valor shaply e  $i$  é o jogador em questão.

FIGURA 29 – MÉDIA DAS DIFERENÇAS DAS CONTRIBUIÇÕES DAS COMBINAÇÕES COM E SEM JOGADOR 1



FONTE: Fonte: o autor  
 LEGENDA: Diferenças nos valores produzidos com e sem o jogador 1

Lundberg e Lee utilizaram este conceito dentro do problema de explicabilidade de modelos de ML e DL, observando como características individuais contribuía para o resultado de um modelo. Para isto, os autores consideraram um conjunto de entradas de um modelo  $x$  e um modelo  $f(x)$ , de modo que fosse possível construir um conjunto  $x'$ , como um vetor binário, que permitisse a exclusão ou não de certa característica, e um modelo explicatório  $g(x')$ , atendendo a duas condições (LUNDBERG; LEE, 2017):

1. Se  $x \approx x'$ , então  $f(x) \approx g(x')$
2. A forma de  $g(x')$  deve ser:

$$g(x') = \phi_0 + \sum_1^N \phi_i x'_i \quad (3.30)$$

Em que  $\phi_0$  é chamado de saída nula do modelo, isto é, a saída média do modelo,  $\phi_i$  é o quanto a características  $i$  altera o resultado do modelo, isto é, o valor *shapley* da características  $i$  (chamado de atribuição) e  $N$  é o total de características.

Atendendo a estas condições, obtém-se uma função explicatória com atribuição aditiva de características. Esta função facilita a explicabilidade do modelo  $f(x)$ , pois observando a equação EQUAÇÃO 3.30, é possível medir o grau de impacto de cada característica no modelo. Lundberg e Lee ainda definem três propriedades desejáveis para a função explicatória (LUNDBERG; LEE, 2017):

1. Acuracidade local, consiste em obter uma função  $g(x')$  com as condições descritas anteriormente.
2. Impacto causado pela falta, isto é, se uma características for excluída do modelo, então sua atribuição deve ser zero (se  $x'_i = 0 \Rightarrow \phi_i = 0$ ). Dessa forma, a única coisa que pode afetar o resultado da função explicatória é a adição de características e não exclusão.

3. Consistência, se a contribuição de uma características muda em um modelo, este efeito não pode ter uma direção contrária na função explicatória. Por exemplo, se em um novo modelo uma determinada características tem maior poder de contribuição, a atribuição desta características na função explicatória não pode diminuir.

Lundberg e Lee (2017) argumentam que apenas os valores SHAP são capazes de atender a todas estas propriedades. No entanto, existe um problema computacional no cálculo dos valores *shapley*, uma vez que é necessário calcular todas as combinações de características, o que pode tornar inviável para a maioria dos modelos (para um conjunto de apenas 32 características, são mais de 17 bilhões de combinações possíveis) (HART, 1989). Para resolver este problema, os autores propuseram alguns tipos de *kernel* para calcular um valor aproximado, dependendo do tipo de modelo e de problema (LUNDBERG; LEE, 2017).

Neste trabalho, foi utilizado o Kernel SHAP, que é um *kernel* para uso mais universal, podendo ser utilizado em todos os tipos de modelo. De forma simplificada, a partir de uma amostra de dados de entrada, o *kernel* calcula uma regressão linear de forma que os coeficientes da regressão se aproximem dos valores *shapley* (LUNDBERG; LEE, 2017).

Exemplos de trabalhos que utilizam valores SHAP incluem prevenção de hipoxemia durante cirurgias (LUNDBERG et al., 2018), análise do processamento de sinais de ECG (ANAND et al., 2022b), análise de séries financeiras (MOKHTARI et al., 2019), previsão de acidentes de caminhões (YANG et al., 2021).

### 3.7 MODELOS DE APRENDIZADO PROFUNDO

Na subseção 3.3.4 é discutido a estrutura básica de uma rede neural artificial. Modelos de DL utilizam dessa estrutura de complexos circuitos algébricos, utilizando, normalmente, em uma grande quantidade de camadas, o qual o termo profundo (ou *deep* em inglês) faz referência. DL é hoje utilizado em diversos tipos de aplicações, como reconhecimento visual de objetos, tradução de idiomas, transcrição de áudios (comumente referido pelo termo em inglês, *speech to text*), síntese de voz/discurso (*text to speech* em inglês), geração de imagens e vídeos, entre outros (RUSSELL; NORVIG, 2021).

Modelos de DL são construídos em blocos ou camadas, de forma similar ao MLP descrito na subseção 3.3.4. Diferentes tipos de blocos podem ser adicionados a rede, cujos propósitos variam desde extrair características, normalizar os dados de saída, retroalimentar as entradas do bloco ou aplicar algum tipo de transformação nos dados. Estas camadas são empilhadas a formar uma rede e, normalmente, ligadas a

uma ou mais camadas de perceptrons com diferentes tamanhos (na nomenclatura de DL, é comum se referir ao perceptron como unidade), totalmente conectadas entre si e por isso recebendo o nome de camada densa (SHAMSHIRBAND et al., 2021).

A camada densa é então conectada a camada de saída, similar ao que ocorre em uma ANN, variando o tipo dependendo da natureza do problema (classificação, regressão ou outro). Esta combinação de camadas é chamada de arquitetura da rede (CHAUHAN; SINGH, 2018).

Treinar modelos de DL é uma atividade desafiadora. Alguns modelos requerem uma grande quantidade de dados para conseguir capturar os padrões necessários, o que além de ter uma base de dados com este volume, também cria desafios computacionais para o treinamento. Outros modelos são de difícil sintonia de pesos, com problemas de gradiente desaparecendo ou explodindo. Estes problemas acontecem devido a instabilidades durante a propagação dos erros, cálculos dos pesos e do gradiente. Mais detalhes destes problemas podem ser encontradas no trabalho de Glorot e Bengio (2010). Existem diferentes estratégias para trabalhar com estes desafios, desde de ferramentas computacionais, como pacotes de desenvolvimento e hardware específico para treinamento de DL, até abordagens para construção dos modelos e ações durante o treinamento, como inicializações pré-definidas, adição de camadas de normalizações e reuso de camadas ou redes já treinadas (CHAUHAN; SINGH, 2018; LAKSHMANAN et al., 2020).

Embora existam estas inúmeras aplicações com modelos de DL, esta seção aborda os tipos de modelos aplicados neste trabalho, redes convolucionais e redes recorrentes. É recomendado ao leitor uma leitura prévia da subseção 3.3.4. A próxima seção aborda modelos de CNN e a seção seguinte os modelos de LSTM.

### 3.7.1 Redes neurais convolucionais

Os estudos envolvendo as CNN surgiram nos anos de 1980 dentro do campo de reconhecimento de imagem e de estudos do córtex cerebral ligado a visão. Um importante avanço aconteceu em 1998 com o trabalho de Lecun et. al. (LECUN et al., 1998), que introduziu uma arquitetura amplamente utilizada para reconhecimento de escrita à mão. Entre as camadas que compõe esta arquitetura, duas camadas tiveram maior destaque: a camada convolucional e a de agrupamento máximo (*max pooling* em inglês) (GÉRON, 2019).

A camada convolucional é a principal camada de uma CNN e responsável por aplicar a operação de convolução nos dados de entrada (dentro da terminologia utilizada em processamento de sinais, a operação aplicada por uma CNN é chamada de convolução cruzada, mas dentro da área de DL convencionou-se a chamar apenas de convolução) com um padrão de pesos de tamanho definido chamado de *kernel*

ou filtro (RUSSELL; NORVIG, 2021). A convolução é uma operação matemática que combina duas formas (ou sinais) em uma terceira forma. Seja os dados de entrada o vetor  $x$  de tamanho  $n$ , considerando que os dados de entrada se originam de uma imagem, então  $n$  corresponde ao total de pixels em uma dimensão e cada entrada no vetor  $x$  é o valor de um pixel. Este valor costuma variar entre 0 e 255 e pode ser, por exemplo, o valor de saturação da cor vermelha em um dado píxel. Seja também o vetor filtro  $k$  de tamanho  $l$ , a convolução é representada pelo símbolo  $*$ , então  $z = x * k$  é definido na 3.31 (ALZUBAIDI et al., 2021).

$$z_i = \sum_{j=1}^l k_j x_{(j+i-(l+1)/2)} \quad (3.31)$$

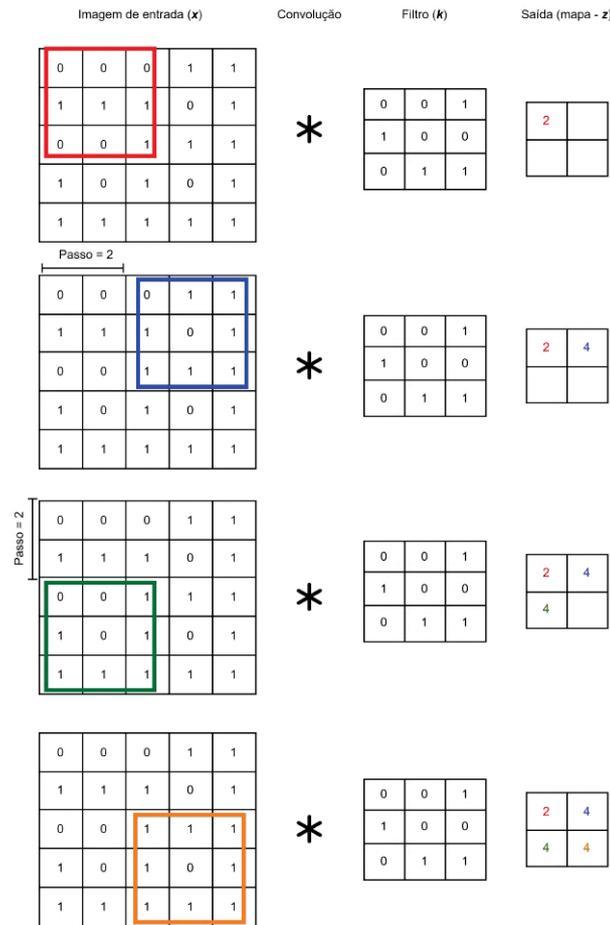
Em outras palavras, para cada posição  $i$ , a equação 3.31 calcula o produto escalar entre o filtro  $k$  e um corte da entrada  $x$  centrado em  $x_i$  e com largura  $l$ . Estes cortes são tomados com base em um hiperparâmetro chamado de passo (ou *stride* em inglês). A camada convolucional aplica diversos filtros (sendo o total um hiperparâmetro definido), aplicando a operação para todos os cortes da imagem. A saída de cada filtro é um vetor  $z$  que compõe o chamado mapa de características (*feature map* em inglês). De forma simplificada, cada mapa no mapa de características torna a imagem menor com base no tamanho do passo (ALZUBAIDI et al., 2021; GUPTA et al., 2022).

A FIGURA 30 apresenta um exemplo de operação de convolução. Nesta FIGURA, a entrada é uma imagem representada por uma matriz 5x5 de 0 e 1, que pode representar uma imagem preto e branco, por exemplo, um filtro 3x3 e passo de tamanho 2. A operação de convolução é mostrada em quatro etapas. Na primeira etapa, representada na primeira linha da FIGURA, um retângulo vermelho destaca a região da entrada que será convolucionada com o filtro. Este retângulo possui o mesmo tamanho do filtro. Dentro deste retângulo, os elementos da primeira linha são analisados com os elementos da primeira linha do filtro. O primeiro zero do retângulo é multiplicado pelo primeiro zero do filtro, o segundo zero do retângulo é multiplicado pelo segundo zero do filtro e assim sucessivamente para todas as linhas. O resultado destas multiplicações é somado e representa o elemento da primeira linha e da primeira coluna da saída (ou mapa), no caso 2 indicado em vermelho.

Este procedimento é repetido para as demais linhas da FIGURA 30. Na segunda linha, o resultado do retângulo azul é 4. É importante observar que a distância do retângulo vermelho para o retângulo azul é de 2 unidades, o tamanho do passo. Na terceira linha, o resultado do retângulo verde é 4 e, finalmente, na quarta linha o retângulo laranja também resulta em 4.

A camada de agrupamento (*pooling*) é responsável adicionar uma variância

FIGURA 30 – EXEMPLO DE OPERAÇÃO NA CAMADA DE CONVOLUÇÃO



FONTE: Fonte: o autor

LEGENDA: Exemplo de uma operação de convolução para uma entrada 5x5 com um filtro 3x3 e passo de tamanho 2

espacial no mapa de características, resumindo por meio de algum procedimento a informação adjacente nos mapas. Esta camada funciona de forma similar à camada convolucional, é aplicado uma operação ao longo de regiões dos mapas com um *kernel* ou filtro de tamanho e passo pré determinados. Esta operação pode variar, dois exemplos comumente utilizados são (ALZUBAIDI et al., 2021):

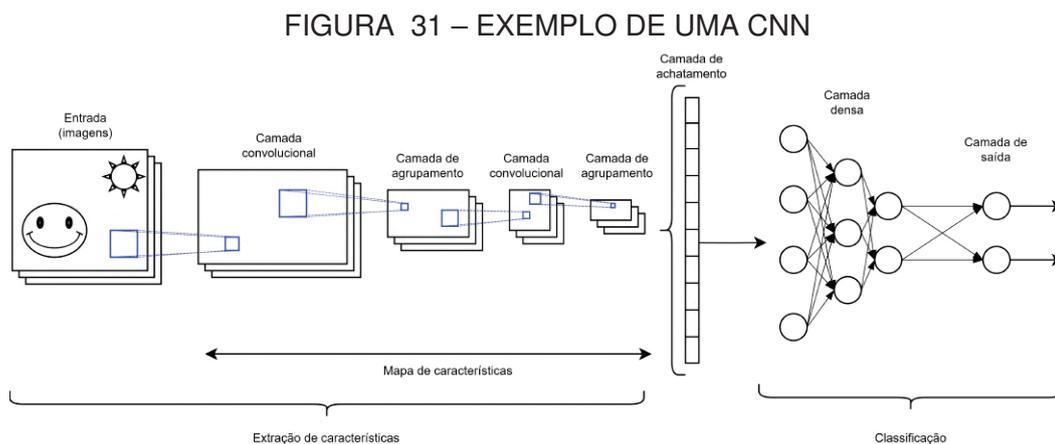
- Agrupamento médio (*average pooling* em inglês), em que o valor médio da região específica é selecionado.
- Agrupamento máximo (*max pooling* em inglês), em que o valor máximo da região é selecionado.

A camada de agrupamento ajuda a remover informação desnecessária e o tamanho dos mapas, mantendo as características mais importantes. Esta operação também ajuda evitar o superajustamento (*overfitting*). Assim como a camada convoluci-

onal, diferentes filtros são aplicados em cada região em cada um dos mapas (GUPTA et al., 2022).

Após o processo de agrupamento, o mapa de características é conectado a uma camada de achatamento (*flattening* em inglês), que tem como objetivo "achatar" os dados, de modo que os mapas são transformados para um formato de vetor compatível com a entrada de uma ANN. Esta próxima camada, composta de uma ANN totalmente conectada, é chamada de camada densa e tem como principal objetivo separar as classes. O resultado desta camada é, então, passado para a camada de saída, que normalmente é construída com uma unidade para cada classe. Desta forma, a CNN tem capacidade extrair as próprias características e depois utiliza-las para classificação (ALZUBAIDI et al., 2021; GUPTA et al., 2022; GÉRON, 2019)

A FIGURA 31 ilustra um exemplo de uma rede CNN completa. Neste exemplo um conjunto de imagens é a entrada de uma camada convolucional que está ligada a uma camada de agrupamento. Os resultados passam por uma segunda camada de convolução, também ligada a uma camada de agrupamento. Os dados passam pela camada de achatamento e vão para camada densa e de saída para classificação. Na FIGURA, os mapas de características e as imagens de entrada são representados como pilhas de matrizes, enquanto a camada de achatamento é representada como um vetor unidimensional. Os retângulos em azul representam regiões das imagens e dos mapas com as regiões do resultado após aplicação das operações de convolução e de achatamento.



FONTE: Fonte: adaptada de (GÉRON, 2019)

LEGENDA: Exemplo de uma CNN com duas camadas de convolução e de agrupamento intercaladas, camada de achatamento e a camada densa

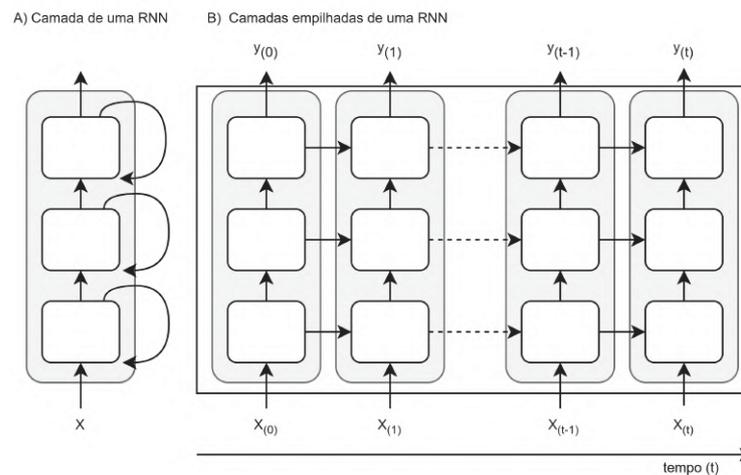
Conforme mencionado, CNN são amplamente utilizadas para reconhecimento de imagens. Além desta aplicação, CNN foram empregadas em reconhecimento de idiomas (LEI et al., 2014), análises histopatológicas (HLAVCHEVA et al., 2019), na análise e detecção de doenças relacionadas à plantas (SHRADHA VERMA; SINGH,

2020), entre outros.

### 3.7.2 Redes neurais recorrentes do tipo *long-short term memory*

As RNN apresentam uma característica diferente das demais redes apresentadas até o momento, a capacidade de retroalimentação. Este mecanismo de retroalimentação permite a criação de um estado interno de memória, isto é, a saída processada alguns momentos antes afeta a saída atual da rede. Dentro da rede, este mecanismo é chamado de célula de memória (YU et al., 2019; GÉRON, 2019). Um exemplo de RNN é mostrado na FIGURA 32. Em "A" é mostrado uma camada cujo as células se retroalimentam e em "B" é mostrado as camadas empilhadas ao longo do tempo. Neste esquema, a cada iteração de unidade temporal, a camada recebe um novo elemento dos dados de entrada  $X$ , indicada por  $x(0)$ ,  $x(1)$ ,  $x(t-1)$  e  $x(t)$ , mas com cada célula recebendo o valor do resultado da iteração passada. Da mesma forma, a saída  $y$  é composta por cada iteração temporal  $y(0)$ ,  $y(1)$ ,  $y(t-1)$  e  $y(t)$ .

FIGURA 32 – EXEMPLO DE UMA RNN

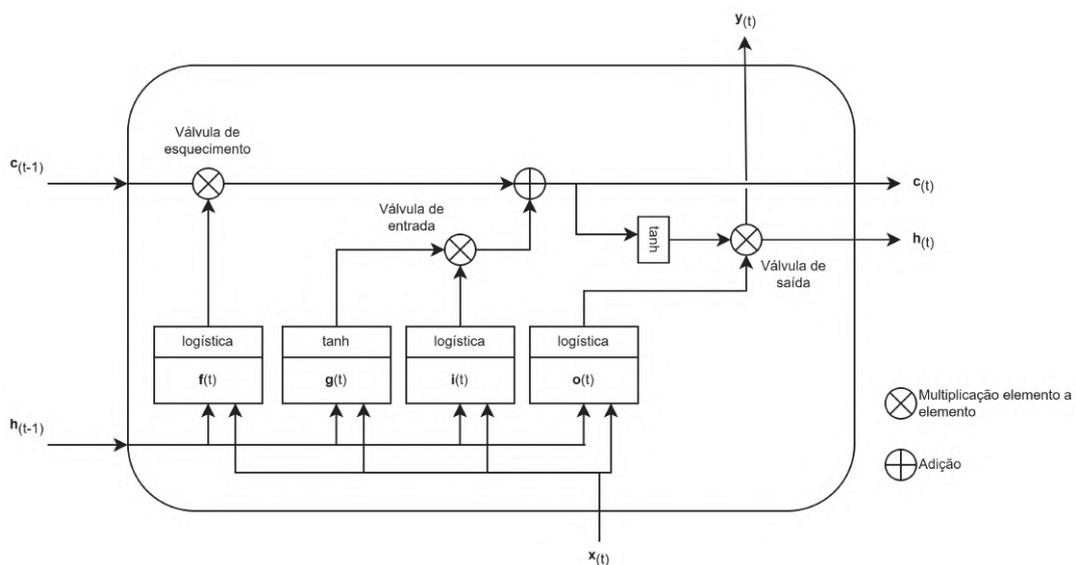


FONTE: Fonte: adaptada de (GÉRON, 2019)  
 LEGENDA: Exemplo de uma RNN, com uma camada com unidades retroalimentadas na esquerda e a propagação temporal das camadas na direita.

Existem diversas implementações de células de memórias e de arquiteturas de RNN. Esta seção foca na arquitetura de LSTM, originalmente proposta em 1997 por Hochreiter e Schmidhuber (HOCHREITER; SCHMIDHUBER, 1997). Esta arquitetura utiliza um tipo específico de célula de memória, que modela uma memória de curto e outra de longo prazo, representado na FIGURA 33. Na FIGURA, é possível observar que a unidade de processamento de LSTM é composta de válvulas que modelam o comportamento de aquisição de memórias de curto e longo prazo e de esquecimento. A memória de curto prazo é representada pelo vetor  $c(t)$ , enquanto que a de longo prazo é representado pelo vetor  $h(t)$ . Estes valores são propagados pela rede a cada

iteração temporal. Na FIGURA,  $y(t)$  é a saída da unidade, enquanto  $x(t)$  é a entrada. O vetor  $f(t)$  controla o acionamento ou não da válvula de esquecimento, que pode zerar o valor da memória de curto prazo ( $c(t)$ ). De forma análoga, o vetor  $i(t)$  é responsável por controlar a válvula de entrada (também chamada de válvula de memória ou de lembrança), que permite a adição ou não de elementos à memória de curto prazo  $c(t)$ . A principal saída da célula é vetor  $g(t)$ . Finalmente, o vetor  $o(t)$  controla a válvula de saída, que determina o quanto da memória e do resultado da unidade será propagado na memória de longo prazo  $h(t)$  (YU et al., 2019; SMAGULOVA; JAMES, 2019; GÉRON, 2019).

FIGURA 33 – ARQUITETURA DE UMA UNIDADE DE LSTM



FONTE: Fonte: adaptada de (GÉRON, 2019)  
 LEGENDA: Unidade de processamento de uma LSTM.

De forma resumida, uma unidade de LSTM aprende a reconhecer a importância de uma entrada, adicionando ou não este valor na memória com uso da válvula de entrada. Armazenando nos vetores de memória e apagando quando for necessário. Todo este mecanismo faz com que a rede LSTM seja capaz de reconhecer padrões temporais em longos textos, áudios ou uma série temporal (YANG et al., 2020).

Alguns exemplos de aplicação de LSTM são previsão meteorológica (MIAO et al., 2020), previsão de preços de ações (SETHIA; RAUT, 2019) e na identificação de software malicioso (SUMARSONO; MASTERS, 2023). Além de unidades LSTM, existem outros tipos de unidades que buscam modelar o comportamento temporal e de memória, um outro exemplo é a unidade GRU (do inglês *gated-recurrent unit*, ou unidade recorrente com portões em tradução livre), o leitor interessado em saber mais pode consultar Dey e Salem (2017).

### 3.8 RESUMO

Ferramentas de ML podem ajudar em diferentes tipos de problemas como regressão ou classificação. Para o desenvolvimento de modelos de ML confiáveis e sem vieses, o processo de treinamento e teste do modelo é fundamental. O uso correto das técnicas ajuda a evitar cenários de *overfitting* e *underfitting*. Encontrar os hiperparâmetros ideais em um problema de ML é uma tarefa custosa e muitas vezes manual e empírica. Para ajudar nesta tarefa, novas tecnologias estão sendo desenvolvidas para não somente automatizar como também adicionar uma heurística de busca. Estas ferramentas muitas vezes podem também trazer novas descobertas, como quais hiperparâmetros mais impactaram o treino, no quesito resultado e tempo. Finalmente, entender como o modelo chegou a uma determinada decisão é fundamental para que os analistas possam tirar o máximo de proveito de um modelo ML.

O próximo capítulo aborda a revisão sistemática da literatura para aplicação de ML e DL na classificação da apneia do sono. Muitas das técnicas descritas nessa seção são adotadas em alguns destes trabalhos.

## 4 REVISÃO DA LITERATURA

Este capítulo aborda o processo de revisão sistemática da literatura. São discutidos a definição das linhas de pesquisa, a metodologia de execução, os achados e um resumo dos principais artigos.

### 4.1 REVISÃO SISTEMÁTICA DA LITERATURA

Em uma análise das bases de dados *web of science* (WoS) e Scopus, utilizando três linhas de pesquisa (com os termos utilizados em inglês):

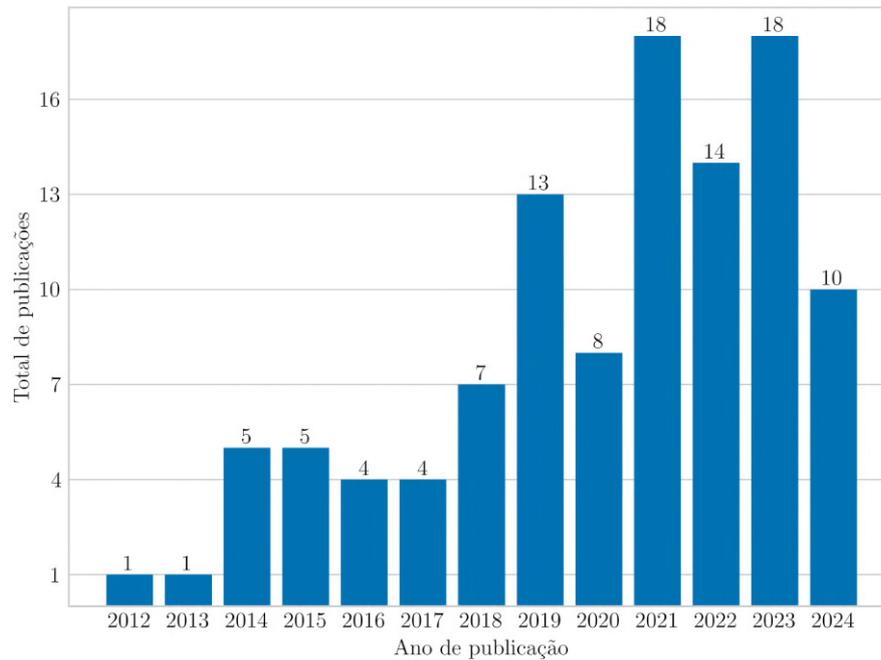
- Linha de pesquisa 1: *sleep apnea*;
- Linha de pesquisa 2: *deep learning OR machine learning*;
- Linha de pesquisa 3: *detection OR classification*.

As linhas de pesquisa foram aplicadas utilizando o operador "E" (*and*) com um filtro no ano de publicação do artigo, em que foram selecionados artigos de 2010 até julho de 2024, e outro no tipo de artigo, selecionando artigos de congresso ou conferências e de periódicos. Nesta etapa, foram encontrados um total de 210 artigos na base Scopus e 246 artigos na base WoS. Com este total de artigos, uma nova etapa de filtragem foi aplicada, os artigos que continham as palavras "*electrocardiogram*" (inglês para eletrocardiograma) ou "ECG" ou "EKG" (acrônimo utilizado em inglês) no título ou resumo.

Com isto, foram obtidos 108 artigos na base Scopus e 63 na WoS. As FIGURAS 34 e 35 mostram a quantidade de publicações por ano. É possível notar que em ambos os casos existe uma tendência de crescimento, apesar de alguns anos em que o número de publicações diminuiu.

Ao todo, estes 171 artigos passaram por dois processos de leitura, o primeiro de títulos e o segundo de resumos. Foram eliminados artigos duplicados, que não utilizavam sinais ECG de canal único, cujos sinais foram coletados de dispositivos *wearables* e que não aplicavam técnicas de ML ou DL, com a exceção para artigos de revisão sistêmica, os quais foram mantidos. Ao final destes processos, um total de 26 artigos foram selecionados para leitura completa, sendo 22 de novos trabalhos e 4 de revisões sistemáticas até o período de julho de 2022. A TABELA 5, TABELA 6 TABELA 7 mostram os 22 artigos de trabalhos novos selecionados e detalham as

FIGURA 34 – BASE SCOPUS



FONTE: Próprio autor

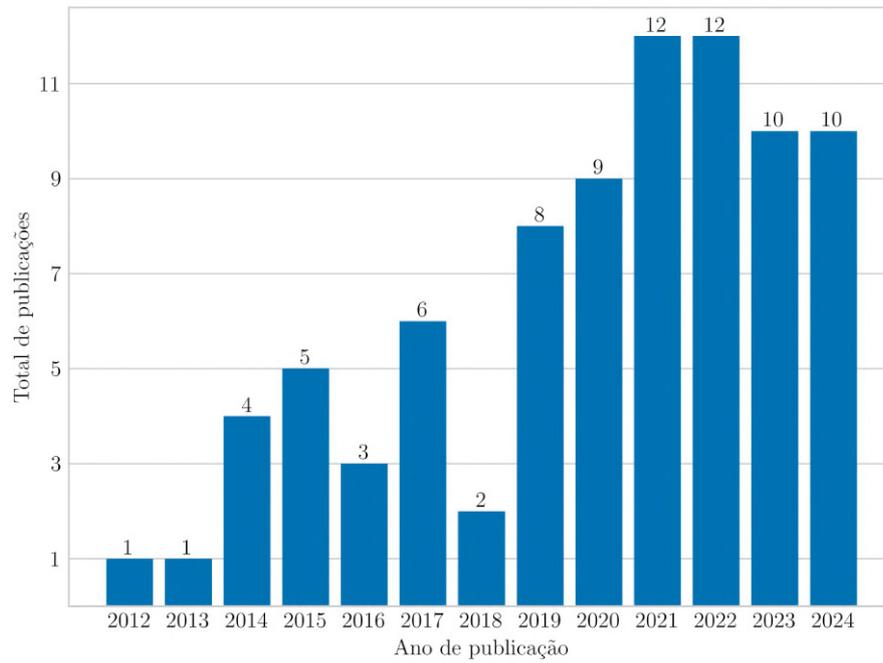
LEGENDA: Total de artigos publicados por ano na base Scopus

bases de dados utilizadas, modelos empregados e parte das metodologias aplicadas nos respectivos trabalhos, como se utilizou de séries R-R ou transformada *wavelet*, quais filtros são aplicados. As séries R-R é composta pela diferença temporal entre os picos da onda R do ECG, em outras palavras, uma série contendo os intervalos da onda R (HAMILTON, 2002). Enquanto a transformada *wavelet* é exemplos de processamento de sinais aplicado ao ECG que permite decompor o sinal em diferentes escalas de tempo e frequência (BALCI et al., 2022). Para o período posterior de julho de 2022 e até julho de 2024, foram selecionados 11 artigos de novos trabalho e 1 de revisão.

Os artigos estudados processaram o ECG de diferentes formas, a maioria utilizou o filtro proposto por (CHEN et al., 2014), referenciado na TABELA como Chen. Este filtro também foi utilizado neste trabalho, conforme será discutido nas seções futuras. Outros autores aplicaram filtros do tipo resposta impulsional infinita (IIR, do inglês *infinite impulse response*), ou do tipo resposta ao impulso finita (FIR, do inglês *finite impulse response*), correções específicas ou outros métodos.

Além dos modelos propostos neste trabalho, no estado da arte também foram utilizados os  $k$  próximos-vizinhos (KNN, do inglês *k-nearest neighbour*), máquinas de aprendizado extremo (ELM, do inglês *extreme learning machines*), *bag of features* (BoF), redes residuais (ResNet, do inglês *residual networks*), redes neurais artificiais

FIGURA 35 – BASE WOS



FONTE: Próprio autor

LEGENDA: Total de artigos publicados por ano na base WoS

(ANN, do inglês *artificial neural networks*), *latent Dirichlet allocation* (LDA), modelo de Monte Carlo cinético (KMC, do inglês *kinetic Monte Carlo*), entre outros. Na próxima seção, é realizada uma síntese destes artigos.

TABELA 5 – RESUMO DOS ARTIGOS SELECIONADOS - PARTE I

Nome do artigo	Base de dados	Séries RR	Wavelet	Pré processamento	Pós processamento
Yang et al. (2022)	Physionet Apnea-ECG; UCDDDB	Sim	Não	Correção específica Filtro passa baixa e passa alta	Padding
Lin et al. (2022)	Physionet Apnea-ECG; Própria	Não	Sim	Ajuste de média zero	N/A
Balci et al. (2022)	Própria	Não	Sim	Filtros específicos para cada sinal	N/A
Bahrami e Forouzanfar (2022)	Physionet Apnea-ECG	Sim	Sim	Filtro Chen	interpolação cúbica (DL)
Srinivasulu et al. (2021)	MIT-BIH	Sim	Não	Não	N/A
Sheta et al. (2021)	Physionet Apnea-ECG	Sim	Não	Filtro Notch	N/A
<b>Almutairi2021 empty citation</b>	Physionet Apnea-ECG	Sim	Não	Filtro Chen	Filtro médio Interpolação cúbica
Faust et al. (2021)	Physionet Apnea-ECG	Sim	Não	Filtro passa alta e filtro passa baixa	N/A
Mukherjee et al. (2021)	Physionet Apnea-ECG	Sim	Não	Método próprio	Normalização Z-score
Feng et al. (2021)	Physionet Apnea-ECG	Sim	Não	Filtro passa alta filtro passa baixa Pan tompkins	Filtro Chen Interpolação cúbica (DL)
Bozkurt et al. (2021)	Própria	Sim	Não	Filtro passa banda Filtro notch Média móvel	N/A
Sharan et al. (2021)	Physionet Apnea-ECG	Sim	Não	N/A	N/A
Fatimah et al. (2020)	Physionet Apnea-ECG; MIT-BIH; UCDDDB	Não	Não	N/A	Remoção de ruído

FONTE: Elaborada pelo autor

TABELA 6 – RESUMO DOS ARTIGOS SELECIONADOS - PARTE II

Nome do artigo	Base de dados	Séries RR	Wavelet	Pré processamento	Pós processamento
Padovano et al. (2020)	Physionet Apnea-ECG	Sim	Não	Butterworth Back-forward IIR Pan tomppkins	Ajuste manual
Ivanko et al. (2020)	Physionet Apnea-ECG	Sim	Não	Wavelet Filtro passa baixa FIR Pan tomppkins	N/A
Erdenebayar et al. (2019)	Própria	Não	Não	Filtro passa-faixa Remoção ruído	N/A
Sadr e Chazal (2019)	Physionet Apnea-ECG	Não	Não	Filtros passa média	N/A
Pinho et al. (2019)	Physionet Apnea-ECG	Sim	Não	Filtro Savitzky-Golay	N/A
Li et al. (2018a)	Physionet Apnea-ECG	Sim	Não	N/A	Filtro Chen
Wongsirichot e Hanskunatai (2017)	Própria PSG	Não	Não	N/A	N/A
Sadr et al. (2015)	Physionet Apnea-ECG	Sim	Não	Filtros passa média	N/A

FONTE: Elaborada pelo autor

TABELA 7 – RESUMO DOS ARTIGOS SELECIONADOS - PARTE III

Nome do artigo	Janela	Modelos	Possui GIT
(YANG et al., 2022)	2 mins	Rede residual espremer e excitar	Não
(LIN et al., 2022)	10 seg e 1 min	BoF, KNN, SVM, ELM	Não
(BALCI et al., 2022)	30 seg	ANN, SVM, RF, NB, KNN, DT, LR	Não
(BAHRAMI; FOROUZANFAR, 2022)	1 min	33 modelos	Sim
(SRINIVASULU et al., 2021)	Registro inteiro	SVM, KNN, Bagging	Não
(SHETA et al., 2021)	1 min	4 modelos DL e 9 ML	Sim
(ALMUTAIRI et al., 2021)	1 min	CNN, LSTM	Não
(FAUST et al., 2021)	100 intervalos RR	LSTM	Não
(MUKHERJEE et al., 2021)	1 min	CNN, LSTM, Ensemble	Sim
(FENG et al., 2021)	1 min	Autoencoders	Não
(BOZKURT et al., 2021)	30 seg	DT, KNN, SVM	Não
(SHARAN et al., 2021)	Registro inteiro	CNN, ResNet	Não
(FATIMAH et al., 2020)	1 min	KNN, SVM, Bootstrap, LogitBoost	Não
(PADOVANO et al., 2020)	1 min	SVM, KNN	Não
(IVANKO et al., 2020)	10 mins	SVM, LR, DT, KNN	Não
(ERDENEBAYAR et al., 2019)	10 seg	CNN, RNN, LSTM	Não
(SADR; CHAZAL, 2019)	1 min	ELM, LDA	Não
(PINHO et al., 2019)	1 min	ANN, SVM	Não
(LI et al., 2018a)	100 intervalos RR(DL)	SVM, ANN, DNN, Autoencoder	Não
(WONGSIRICHOT; HANSKUNATAI, 2017)	N/A	KMC, SVM, kNN, MLP	Não
(SADR et al., 2015)	1 min	ELM	Não

FONTE: Elaborada pelo autor

Como resume a TABELA 5, as mais variadas estratégias de modelagem foram experimentadas na classificação da apneia do sono, tanto utilizando modelos de aprendizado de máquina clássico e de aprendizado profundo. Em geral, a maioria dos trabalhos utilizam bases públicas para construção dos modelos. As principais bases públicas são: a PhysioNet (utilizada neste trabalho), a UCDDDB da Universidade College de Dublin e a MIT-BIH polissonografia do laboratório do sono do hospital Beth Israel de Boston (GOLDBERGER et al., 2000). O processamento do ECG e a extração e cálculo das características também variou nos trabalhos, com alguns trabalhos extraíndo as séries R-R, enquanto outros calculando a transformada de wavelet. Também houveram trabalhos que treinaram os modelos diretamente no ECG.

O processo de filtragem também variou. Trabalhos como Almutairi et al. (2021) e Feng et al. (2021), por exemplo, aplicaram duas etapas de processamento e filtragem de sinais, outros aplicaram filtros após extraírem as séries R-R, como Mukherjee et al. (2021). O filtro aplicado variou entre filtros passa-baixa, passa-médias e passa-altas. Alguns trabalhos aplicaram interpolação de dados como estratégia de processamento do ECG.

Com o suporte do software VOSViewer, é possível analisar as palavras chaves dos artigos e relacioná-las em um grafo. Cada cluster é conectado com termos vizinhos comumente empregados e sua análise permite uma leitura dos principais termos e suas relações com a base de artigos. O VOSViewer foi utilizado na base da Scopus e da WoS antes da filtragem se o título ou resumo continham o termo "ECG", ou seja quando as bases tinham 210 e 246 artigos, respectivamente. O resultado desta análise é exibida na FIGURA 36 para base Scopus e na FIGURA 37 para base WoS.

Nas FIGURAS 36 e 37 cada ponto é uma palavra-chave são compostas por agrupamentos, em que cada ponto é uma palavra chave. O tamanho do ponto representa o quanto aquela palavra-chave apareceu nos artigos, quanto maior o ponto, mais vezes a palavra apareceu. Na base Scopus, as palavras com mais destaque foram (em tradução livre) "estágio do sono", "doença", "memória de curto prazo", "saturação de oxigênio" e "máquinas de vetores de suporte", com outras em menor destaque. Palavras similares estão em destaque na base WoS, mas também tiveram grande destaque "índice de apneia-hipopneia", "máquinas de vetores suporte" e "espectrograma".

Cada um dos pontos é ligado a outras palavras-chaves que comumente aparecem juntas, formando uma rede, com algumas palavras se comportando com grandes centros de conexão, como o caso de "máquinas de vetores de suporte", em ambas as FIGURAS. Outras estão com menos conexões, um exemplo da base Scopus é a palavra-chave "extração de características" ligada aos termos "máquinas de vetores de suporte", "detecção de apneia do sono", "detecção precoce", "redes residuais", entre outros. É possível analisar que a extração de características está comumente associado

aos sinais fisiológicos e a base de dados utilizada. Um exemplo da base WoS é o termo "aprendizado de máquina extremo" ligado aos termos "PCA", "respiração derivado do ECG", "variabilidade cardíaca", entre outros. Isto pode indicar que estratégias de aprendizado de máquina extremo são comumente empregadas com uso de modelos de KNN e podem ajudar na acurácia da classificação.

Por fim, cada cor nas FIGURAS 36 e 37 é um grupo formado pela proximidade das palavras-chave. Existem 11 grupos na base Scopus e 10 grupos na base WOS, separados por cores diferentes. Na FIGURA da base WoS os grupos azul escuro e verde contém algoritmos de classificação e alguns sinais fisiológicos, sendo que no grupo verde é maior, com mais palavras-chaves ligadas a sinais fisiológicos e base de dados. Os grupos vermelho e azul claro contém palavras predominantemente relacionadas a AOS e a saúde do indivíduo e por fim os grupos amarelo, marrom, roxo, rosa e bege contém palavras-chaves variada, focando em etapas da classificação, métricas de acuracidade e informações sobre AOS e a polissonografia. Os grupos da FIGURA da base Scopus obtiveram um comportamento semelhante, com os grupos verde escuro e azul com as técnicas de classificação, os grupo verde claro e vermelho com termos relacionado à saúde e a AOS, com grupo amarelo. Os grupos azul claro e marrom, contém termos relacionados aos sinais fisiológicos, enquanto os grupos laranja, roxo e rosa um misto de termos de aplicação e métodos, com tratamento, estágios do sono e risco da AOS.





## 4.2 SÍNTESE DAS PRINCIPAIS REFERÊNCIAS

Em Sadr et al. (2015) foi utilizado um comitê de aprendizado extremo para obter uma acuracidade de 82,5% na base do MIT, classificando as séries R-R extraídas em janelas de 1 minutos dos sinais de ECG. O trabalho de Wongsirichot e Hanskunatai (2017) realizou uma análise sobre impacto das características e dos sinais fisiológicos no processo de classificação. Os dados utilizados vieram de exames de polissonografia. De todas os sinais coletados, os autores acharam que o ECG e o sinal de oxigenação foram os mais importantes. O melhor modelo construído foi um KNN com 95,2% de acuracidade. Para endereçar o problema de extração de características, Li et al. (2018a) utilizou uma abordagem não supervisionada para extrair as características e classificar os sinais de ECG, com um modelo de cadeia de Markov em uma rede profunda, de modo que os sinais passavam por um processo de auto-codificação (*autoencoders* em inglês). Os autores obtiveram um resultado de 85,0% de acuracidade na base da PhysioNet.

O trabalho de Aggarwal et al. (2018) também testou diferentes arquiteturas DL, desenvolvendo um ambiente computacional para modelagem fim-a-fim de DL. O melhor modelo encontrado obteve uma acuracidade de 74,1%. Diferentes abordagens de DL foram comparadas em Erdenebayar et al. (2019). O melhor resultado encontrado foi de 99,0% de acuracidade com uma rede profunda com unidades recorrentes com portões (GRU, do inglês *gated-recurrent unit*). Os autores utilizaram uma base própria, que coletou o ECG de 86 pacientes. Ainda analisando os efeitos da seleção de características, Pinho et al. (2019), calcularam 50 características da HRV, 34 da EDR e 64 da densidade de potência espectral extraídas de uma transformada rápida de Fourier. Todas estas características foram ranqueadas de acordo com seu poder de discriminação, calculado pelo método Relevância Discriminante. O melhor resultado foi encontrado com um conjunto de 20 características em um modelo de ANN, com 82,1% de acuracidade.

Em Ivanko et al. (2020) foi utilizado a base PhysioNet para analisarem e estudarem os efeitos das características na classificação da AOS. Para isto extraíram janelas de 10 minutos e analisaram 24 características do domínio do tempo, do domínio da frequência e do espectrograma da HRV. Os autores também treinaram uma SVM e os melhores resultados foram com um conjunto de 9 características, com uma acuracidade de 98,7%. No trabalho de Fatimah et al. (2020) também foi analisado o efeito de seleção de características ao classificar a AOS nas bases da PhysioNet, MIT-BIH e UCDDb, aplicando uma decomposição de Fourier para calcular a entropia e a média do desvio padrão absoluto. O melhor modelo foi uma SVM com resultado de 94,7% de acuracidade. Padovano et al. (2020) foi treinado uma SVM e um modelo do tipo KNN, com características calculadas das séries R-R a partir dos dados de ECG da

base da PhysioNet, obtendo 81.4% de acurácia com a melhor SVM.

No trabalho de Feng et al. (2021), técnicas de aprendizado não-supervisionado foram utilizadas. Partindo da base da PhysioNet, extraíndo as séries R-R, os autores desenvolveram um modelo baseado em *autoencoders* esparsos empilhados em domínio de frequência e em tempo-dependentes sensitivos ao custo, alcançando uma acuracidade de 85,1%, com a vantagem de não ser necessário uma prévia classificação dos dados. Já os autores de Sharan et al. (2021) propuseram utilizar o ECG cru, isto é, sem nenhum tipo de processamento. Os autores utilizaram a base da PhysioNet para treinar uma rede profunda residual, um tipo de DL, com um resultado de 93,1% de acuracidade.

Em Sheta et al. (2021) foi proposto um sistema de auxílio diagnóstico baseado em ML e ML, treinando 9 modelos de ML e 4 de ML com a base da PhysioNet, após processar o ECG com um filtro de rejeição de frequência (*notch*) de segunda ordem do tipo IIR e extraíndo um total de nove características das séries R-R. Os hiperparâmetros foram selecionados através do uso de uma ferramenta de otimização do software Matlab para os modelos ML, enquanto para os modelos DL foram configurados hiperparâmetros recomendados em outros trabalhos acadêmicos. O melhor modelo foi um híbrido de CNN com LSTM, que obteve 86,3% de acuracidade, 89,0% de sensibilidade, 87,0% de especificidade e 87,7% de F-1 score. Outro trabalho que obteve o melhor resultado com um híbrido de CNN e LSTM foi Almutairi et al. (2021), que também utilizou a base PhysioNet e usando uma interpolação cúbica de séries R-R com amplitude dos intervalos R-R, obteve 90,9% de acuracidade. Outra abordagem híbrida, realizando uma agregação de técnicas de ML e DL foi proposta em Mukherjee et al. (2021). Os autores utilizaram a base da PhysioNet e combinaram os modelos DL descritos em outros trabalhos e outras técnicas ML, obtendo a melhor acuracidade de 85,6% ao agregar um MLP as arquiteturas de DL.

Outras abordagens também foram testadas, como em Faust et al. (2021) que utilizaram um modelo de DL do tipo rede neural de memória de longo prazo com recorrência LSTM nas série R-R extraídas do ECG da base da PhysioNet e obtiveram uma acurácia de 81,3%. Bozkurt et al. (2021) trabalharam com uma base própria, coletando o ECG de pacientes para o estudo. Em seguida, extraíram as séries R-R, calcularam diversas características e treinaram oito tipos de modelos ML, obtendo uma acuracidade de 85,7% com um *ensemble* de DT, SVM e modelos KNN. Srinivasulu et al. (2021) utilizaram a base MIT-BIH, extraíndo as séries R-R e calculando características a partir delas, para treinar 4 tipos diferentes de ML, com a maior acurácia obtida com um *ensemble-bagged* de DT de 89,6%.

Em Yang et al. (2022), os autores trabalharam com características extraídas da variabilidade da frequência cardíaca e da respiração derivada do ECG, (EDR, do

inglês *ECG derived respiratory*) para treinar uma rede de grupo residual de espremer e excitar unidimensional, um tipo de modelagem DL. As bases utilizadas da PhysioNet e da UCDDDB. Para o processamento do ECG, os autores corrigiram algumas entradas da base PhysioNet manualmente e extraíram janelas de 2 minutos dos ECG. Para cada janela, aplicaram um filtro passa-banda composto de filtros passa-baixa e passa-alta conectados em cascata. Após estas etapas, detectaram os picos R-R com o uso do algoritmo de Hamilton, obtendo assim as séries R-R e elaboraram uma técnica para identificar o complexo QRS, obtendo-se a série de Q-amplitude (QA). De posse da posição dos picos Q, os autores calcularam a EDR a partir do método R-amplitude (RA), e então preencheram (*padding*) três sequências de série R-R, RA e QA para possuírem um mesmo tamanho (requisito de entrada em redes do tipo CNN). Esta entrada foi utilizada para treinar o modelo escolhido. A arquitetura deste modelo alterna camadas de convolução, com função de ativação unidade linear retificada (ReLU, do inglês *rectified linear unit*), com camadas totalmente conectadas (com uso de funções sigmóides como ativação). Os hiperparâmetros foram escolhidos de forma empírica e manual, enquanto se observavam os resultados obtidos durante o treinamento da rede. Os autores obtiveram 90,3% de acuracidade, 87,6% de sensibilidade, 91,9% de especificidade e 0,873 de score F-1 na base PhysioNet. Na base UCDDDB obtiveram 75,1%, 61,1% e 80,8% de acuracidade, sensibilidade e especificidade respectivamente.

Lin et al. (2022) foi explorado o uso do paradigma *bag-of-features* (BoF, "saco de características", em tradução livre) na classificação da AOS. No trabalho, os autores trabalharam com a base da PhysioNet e com uma base própria. Processaram o ECG em duas janelas diferentes, uma com 10 e outra com 60 segundos, e ajustaram os dados de modo que a média do sinal fosse zero. Então calcularam a transformada contínua de *wavelet* dos sinais ECG. A técnica BoF foi então utilizada para marcar e extrair nos espectrogramas elementos chaves que ajudam a discriminar respirações com AOS de respirações normais. Estes elementos foram, então, utilizados para treinar três modelos de ML: SVM, KNN e bagging de DT e de KNN. Os hiperparâmetros foram escolhidos através de uma busca em grade com validação cruzada de 5-fold. O melhor modelo na base PhysioNet foi o bagging, com desempenho de 91,4% de acuracidade, 89,8% de sensibilidade e 92,4% de especificidade. Em Balci et al. (2022), os autores utilizaram diversos sinais fisiológicos para avaliar diferentes modelos de ML na classificação da apneia. Os autores utilizaram uma base própria, que consistia nos sinais de ECG, pressão arterial, ronco de pressão, saturação de oxigênio, pulso e sinais do tórax medidos em exames de polissonografia de 19 pacientes. Todos os sinais foram pré-processados com a remoção de suas componentes contínuas (DC). Então cada sinal passou por um processamento diferente. O sinal de ECG foram selecionadas frequências entre 0,05 Hz e 40 Hz; para pressão de ronco, foi aplicado um filtro passa alta de 10 Hz; no sinal SpO2 foi processado em um filtro passa alta de 0,5 Hz; para o

signal de pulso um filtro passa banda foi aplicado, com frequências de corte entre 0,5 Hz e 2 Hz e, por fim, no sinal do tórax foi aplicado um filtro passa-baixa de 0,5 Hz. Após o processamento dos sinais, os sinais foram segmentados em janelas de 30 segundos e 35 características do domínio do tempo e da frequência, que foram utilizadas para treinar os modelos ANN, SVM, RF, Naive-Bayes, KNN, DT e LR. Os hiperparâmetros foram definidos previamente e fixados. O melhor modelo foi o RF com uma acuracidade de 76,3%, sensibilidade de 76,3%, 73,2% de especificidade e F1 score de 0,725.

No trabalho de Bahrami e Forouzanfar (2022), os autores compararam um total de 33 modelos de ML e de DL treinados com a base da PhysioNet. Para o processamento dos dados, os autores extraíram as séries R-R e aplicaram o filtro passa-média proposto por Chen et al. (2014) e segmentaram os sinais em janelas de 1 minuto. Para os modelos de ML, foram extraídas um total de 27 características do domínio do tempo e da frequência da HRV. Os autores, então, aplicaram a técnica de redução de dimensionalidade PCA e treinaram os modelos com uma validação cruzado com 5-fold. Os hiperparâmetros foram selecionados com base na validação do melhor modelo. O melhor ML foi o VM, com uma acuracidade de 79,4%, sensibilidade de 68,7%, especificidade de 85,9% e F1 score de 71,8%. Os autores ainda analisaram a importância de cada características no processo decisório dos modelos, para isto treinaram um modelo árvores extras (*extra tree*), encontrando as características frequenciais como as mais importantes, com destaque para HF e VLF. Nas características temporais, a mais significativa foi a SD.

Os modelos de DL foram treinados com as séries R-R interpoladas com um sinal cúbico de 3 Hz e série de amplitudes dos R-picos. Foram treinados modelos do tipo CNN, RNN e híbridos. Para o componente temporal dos modelos tipo RNN, os dados de entrada foram segmentados em 2 segmentos de 30 segundos cada. A maior acuracidade foi de 88.1% pelo modelo híbrido ZFNet-BiLSTM, que obteve uma sensibilidade de 81,5%, uma especificidade de 92,3% e um F1 score de 84,0%.

Em Ayashm et al. (2022), os autores utilizaram tanto os dados da base da Physionet, quanto os dados de uma base própria. Uma CNN totalmente conectada foi treinada diretamente com os sinais do ECG filtrados. A CNN classificou cada segmento de 60 segundos e, ao final da rede, as classificações passaram por uma camada global de agrupamento, que calculou as médias das classificações para a predição final. Os autores obtiveram uma acurácia de 98,6%, uma sensibilidade de 98,7% e uma especificidade de 98,6%. No trabalho de Ayatollahi et al. (2022), os sinais de ECG da base de dados da Physionet foram segmentados em janelas de 1 minuto, em que um filtro de terceira ordem de Butterworth foi aplicado e transformados em sinais bidimensionais com uso do algoritmo de gráfico de recorrência. Os sinais foram utilizados para finalizar o treinamento de um conjunto de CNN pré-treinadas. Os

melhores resultados obtidos alcançaram uma acurácia de 93,3%, uma precisão de 92,1%, um *recall* de 93,3% e um F1-score de 92,5%.

Yeo et al. (2022) foi utilizado a base de polissonografia do *National Sleep Research Resource*, em que os sinais de ECG foram processados para extração das séries R-R e sinais de derivação respiratória. Estes sinais foram utilizados para treinar três diferentes tipos de redes residuais convolucionais. Os modelos classificaram tanto a ocorrência ou não de eventos de AOS, quanto estimaram o IAH. Para a classificação de ocorrência de AOS, os autores obtiveram 86,0% de acuracidade, 85,0% de especificidade e 86,0% de sensibilidade.

Em Liu et al. (2023), os autores utilizaram uma estratégia de modelagem mista, em que combinaram uma rede CNN com uma modelagem por sequência utilizando *encoder* e *decoder* para prever a ocorrência de eventos de AOS na base da PhysioNet. Os sinais de ECG foram filtrados com um filtro passa faixa do tipo Butterworth e segmentados de duas formas diferentes, em janelas de um e três minutos. Os melhores resultados foram obtidos com a janela de 3 minutos, em que foi obtido uma acurácia de 88,2%, uma sensibilidade de 78,5%, uma especificidade de 94,1% e um F1-score de 0,89. No trabalho de Srivastava et al. (2023), os autores treinaram diferentes tipos de modelos de DL, variando entre redes residuais, CNN, LSTM, redes totalmente conectadas e abordagens mistas. Os modelos foram treinados com os sinais das séries R-R, separados em janelas de 1 minuto. Estes sinais foram extraídos dos ECGs filtrados por um filtro passa-média dos da base de dados da PhysioNet. O melhor modelo obtido foi com uma rede residual com uma acuracidade de 90,1%.

Os autores de Thompson et al. (2023) treinaram uma CNN com uma camada convolucional e compararam os resultados com outros tipos de modelos de ML. Os autores utilizaram a base de dados da PhysioNet para criar diferentes conjuntos de treinamento variando a janela de pontos em cada conjunto. Os melhores resultados foram encontrados com uma janela de 500 pontos de dados para o modelo CNN, com uma acuracidade de 97,0%, 97,0% de especificidade e 97,4% de sensibilidade. Em Garcia-Vicente et al. (2023) utilizaram uma base de dados provida pelo *National Sleep Research Resource* para classificação de eventos de AOS em crianças. Os sinais do ECG passaram por dois processo de filtragem, primeiro por um filtro com uma componente contínua e seguido por um filtro passa baixa. Os sinais foram divididos em janelas de 10 minutos. Uma CNN foi treinada para estimar a quantidade de eventos de AOS dentro da janela de 10 minutos, calculando uma estimativa para o IAH. Os autores obtiveram uma acuracidade de 75,9% para janelas com 1 evento por hora, 87,0% para janelas com 5 eventos por hora e 92,0% para janelas com 10 eventos por hora.

No trabalho Lin et al. (2024), os autores aplicaram a transformada de *wavelet*

nos sinais do ECG da base da PhysioNet para treinar uma CNN. Os sinais foram segmentados em janelas de 1 minuto e filtrados com um filtro de Butterworth. Então, foram gerados imagens de tempo-frequência com a transformada de *wavelet* em três janelas temporais diferentes, de 1, 2 e 3 minutos. Os melhores resultados encontrados foram com janela de 3 minutos, em que foi obtido uma acuracidade de 92,5%, uma sensibilidade de 84,5%, uma especificidade de 97,3% e um F1-score de 0,89.

Em Liu et al. (2024), os autores utilizaram uma base de dados própria e a base da PhysioNet para treinar um modelo de CNN e um modelo híbrido de CNN com XgBoost, obtendo uma acuracidade de 92,8%. Os autores do trabalho Wang et al. (2024) também utilizaram uma base de dados própria e a base da PhysioNet em uma estratégia híbrida de modelagem, combinando modelos generativos, métodos *fuzzy* e modelos de CNN. Os sinais de ECG foram filtrados com um filtro de Butterworth, seguido por um filtro passa média. Foram extraídas as séries R-R e as séries de amplitude R. O modelo generativo ajudou a criar dados artificiais para aumentar a base de dados, estes dados foram posteriormente filtrados com uso de métodos *fuzzy*. Os dados foram utilizados para treinar uma rede híbrida com camadas residuais e convolucionais. Os autores obtiveram uma acurácia de 94,3% para base da PhysioNet e 90,0% para base própria.

#### 4.3 RESUMO

Este capítulo apresentou a revisão sistemática da literatura para classificação da AOS com uso de ML e DL. A revisão avaliou artigos de 2010 até julho de 2022, selecionando 22 artigos para estudo detalhado. Estes 22 artigos foram então apresentados, discutindo alguns detalhes de metodologia, como quais bases de dados foram utilizadas, quais os modelos de ML e DL foram empregados, como os hiperparâmetros foram sintonizadas, entre outros aspectos. Os artigos são resumidos, contextualizando melhor o estado da arte.

## **Parte II**

### **Materiais e métodos**

## 5 METODOLOGIA

Neste capítulo é detalhado os materiais e métodos empregados no desenvolvimento da pesquisa. A primeira seção aborda a base pública PhysioNet, apresentando os dados, contexto de coleta e detalhando alguns aspectos; a seção seguinte mostra o macro fluxo de trabalho para os modelos de ML e DL. As seções subsequentes entram em cada passo deste macro fluxo detalhando: extração das características, escolha dos hiperparâmetros, redução da dimensionalidade, treinamento e teste dos modelos, análise de resultados, entre outros. Todo projeto foi implementado na linguagem de programação *Python*. Para os modelos de ML, foi utilizado a versão 3.7.6 do *Python* com compilador MSC (acrônimo para *Microsoft C*) versão 1916 com uma arquitetura de 64 bit (AMD64, em que AMD é um acrônimo para *Advanced Micro Devices*) em um computador *desktop* com *Windows 10 Pro*, com processador Intel Core i5-9400F de 2,90GHz, 16 GB de memória RAM (do inglês *random access memory*) e placa gráfica NVidia GTX 1650 com 16 GB de vRAM (do inglês *video random access memory*), em ambiente Jupyter Notebook.

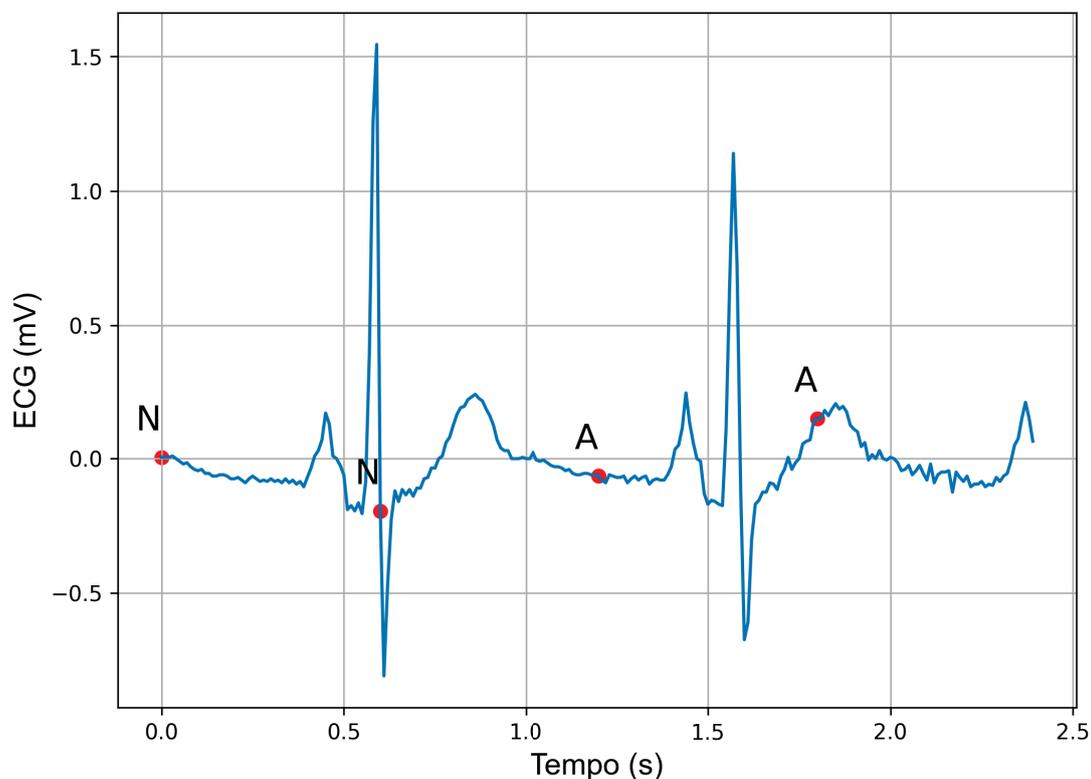
Para os modelos de DL, foi utilizado a versão 3.10 do *Python* com um compilador GCC (acrônimo para GNU *Compiler Collection*, ou coleção de compiladores do projeto GNU, em tradução livre, e GNU é um acrônimo recursivo para *GNU's Not Unix!*) versão 11.4.0 e foram executados em ambiente de nuvem Google Colab Pro+ (versão paga) com processamento em uma placa gráfica NVIDIA A100-SXM4-40GB. Foi utilizada a versão paga uma vez que a versão gratuita possui limitações de tempo de execução que não atendiam aos tempos de execução dos experimentos desta dissertação. Na data da execução dos experimentos desta dissertação o Google Colab Pro+ custava R\$ 258,00 mensais.

### 5.1 BASE DE DADOS

A base de dados utilizada foi a Apnea-ECG Database 1.0.0 da PhysioNet. A base consiste em 70 registros do ECG de canal único, gravados a 100 Hz de 32 indivíduos, sendo 25 homens e 7 mulheres. Os registros variam entre 401 a 534 minutos e foram analisados por especialistas que anotaram em cada minuto se o paciente estava em apneia ou não. Se no início de cada minuto o paciente estava em apneia do sono, aquele minuto foi marcado com o símbolo A, caso contrário com símbolo N. A FIGURA 38 mostra um exemplo de um ECG já processado com registro dos respectivos rótulos. Importante ressaltar que todo o minuto é atribuído aquele rótulo, ou seja, se no início do minuto o paciente estava com apneia, recebeu o rótulo A, e se durante este minuto o paciente teve respirações normais, o rótulo permaneceu

como A.

FIGURA 38 – EXEMPLO DE ECG COM MARCAÇÕES DA BASE DE DADOS



FONTE: Fonte: o autor

LEGENDA: Exemplo de ECG com marcações

Os registros foram divididos em quatro grupos: A, B, C e X. O grupo corresponde a pacientes com índices de apneia do sono elevados (mais de 5 horas), o grupo B são pacientes com ao menos 1 hora de apneia registrada, o grupo C é o grupo controle, com nenhuma hora completa de apneia registrada. O grupo X é o grupo originalmente designado para teste dos modelos, com pacientes que variam em diferentes níveis de AOS. As TABELAS 8, 9, 10, e 11 apresentam alguns indicadores e dados dos pacientes de cada um dos grupos respectivamente. O grau de severidade de apneia foi classificado de acordo com TABELA 2. Ao final de cada TABELA, é apresentado uma média de alguns dados daquele grupo. Na TABELA 8, todos os pacientes são do sexo masculino e possui uma predominância de pacientes com severidade grave para AOS, com apenas dois com severidade moderada e um classificada como leve. Na média do grupo A teve um IAH de 48, ou seja, 48 episódios de AOS por hora. No grupo B (TABELA 9), composto de apenas de cinco pacientes, com 2 possuindo AOS moderada e 3 pacientes, um deles do sexo feminino, com IAH menor que 6 (sem apneia). O grupo

controle C, na TABELA 10, é composto por 4 mulheres e 6 homens, todos eles com IAH zero. Finalmente, no grupo X, designado para teste, possui 35 pacientes, dos quais a maioria é homem, com 14 pacientes com AOS grave, 5 moderada, 4 leve e 12 sem apneia, formando um grupo misto (PENZEL et al., 2000; YANG et al., 2022; ALMUTAIRI et al., 2021; BAHRAMI; FOROUZANFAR, 2022).

TABELA 8 – GRUPO A DE PACIENTES DA BASE DE DADOS

Gravação	Duração (mins)	Minutos sem apneia	Minutos com apneia	IAH	Severidade	Idade	Sexo	Altura (cm)	Peso (kg)
a01	490	20	470	70	Grave	51	M	175	102
a02	529	109	420	70	Grave	38	M	180	120
a03	520	274	246	39	Grave	54	M	168	80
a04	493	40	453	77	Grave	52	M	173	121
a05	455	179	276	41	Grave	58	M	176	78
a06	511	305	206	25	Moderada	63	M	179	104
a07	512	190	322	63	Grave	44	M	177	105
a08	502	313	189	42	Grave	51	M	179	88
a09	496	115	381	32	Grave	52	M	178	82
a10	518	418	100	21	Moderada	58	M	176	78
a11	467	245	222	14	Leve	58	M	168	103
a12	578	44	534	80	Grave	52	M	173	121
a13	496	252	244	42	Grave	51	M	179	88
a14	510	127	383	55	Grave	51	M	175	102
a15	511	143	368	52	Grave	60	M	176	113
a16	483	163	320	41	Grave	44	M	177	105
a17	486	328	158	33	Grave	40	M	179	96
a18	490	52	438	82	Grave	52	M	178	82
a19	503	298	205	34	Grave	55	M	178	90
a20	511	196	315	41	Grave	58	M	176	78
<b>Média</b>	<b>503</b>	<b>191</b>	<b>313</b>	<b>48</b>	<b>N/A</b>	<b>52</b>	<b>N/A</b>	<b>176</b>	<b>97</b>

FONTE: Adaptada de (PENZEL et al., 2000)

TABELA 9 – GRUPO B DE PACIENTES DA BASE DE DADOS

Gravação	Duração (mins)	Minutos sem apneia	Minutos com apneia	IAH	Severidade	Idade	Sexo	Altura (cm)	Peso (kg)
b01	488	469	19	0.24	Sem apneia	44	F	170	63
b02	518	425	93	19	Moderada	53	M	176	85
b03	442	369	73	24	Moderada	53	M	176	85
b04	430	420	10	0.7	Sem apneia	42	M	180	64
b05	434	377	57	5	Sem apneia	52	M	180	135
<b>Média</b>	<b>462</b>	<b>412</b>	<b>50</b>	<b>10</b>	<b>-</b>	<b>49</b>	<b>-</b>	<b>176</b>	<b>86</b>

FONTE: Adaptada de (PENZEL et al., 2000)

TABELA 10 – GRUPO C DE PACIENTES DA BASE DE DADOS

Gravação	Duração (mins)	Minutos sem apneia	Minutos com apneia	IAH	Severidade	Idade	Sexo	Altura (cm)	Peso (kg)
c01	485	485	0	0	Sem apneia	31	M	184	74
c02	503	502	1	0	Sem apneia	37	M	180	83
c03	455	455	0	0	Sem apneia	39	M	184	65
c04	483	483	0	0	Sem apneia	41	F	180	65
c05	467	464	3	0	Sem apneia	28	F	169	57
c06	469	468	1	0	Sem apneia	28	F	171	65
c07	454	450	4	0	Sem apneia	30	F	168	56
c08	535	535	0	0	Sem apneia	42	M	180	64
c09	469	467	2	0	Sem apneia	37	M	180	83
c10	432	431	1	0	Sem apneia	27	M	184	72
<b>Média</b>	<b>475</b>	<b>474</b>	<b>1</b>	<b>0</b>	<b>-</b>	<b>34</b>	<b>-</b>	<b>178</b>	<b>68</b>

FONTE: Adaptada de (PENZEL et al., 2000)

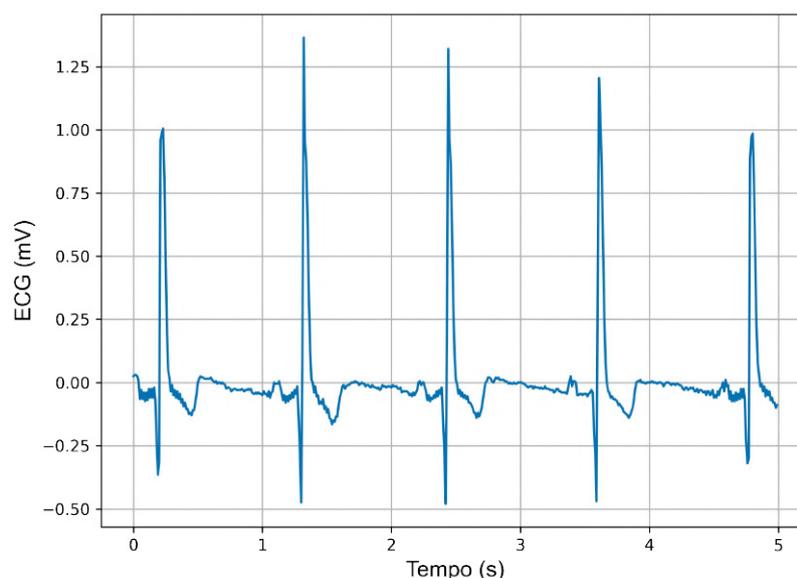


TABELA 11 – GRUPO X DE PACIENTES DA BASE DE DADOS

Gravação	Duração (mins)	Minutos sem apneia	Minutos com apneia	IAH	Severidade	Idade	Sexo	Altura (cm)	Peso (kg)
x01	524	149	375	63	Grave	44	M	177	105
x02	470	261	209	38	Grave	46	M	167	69
x03	466	454	12	0	Sem apneia	44	F	170	63
x04	483	483	0	0	Sem apneia	39	M	184	65
x05	506	190	316	34	Grave	55	M	178	90
x06	451	451	0	0	Sem apneia	31	M	170	66
x07	510	270	240	21	Moderada	58	M	176	78
x08	518	194	324	48	Grave	55	M	178	90
x09	509	342	167	19	Moderada	43	M	177	80
x10	511	415	96	10	Leve	39	M	170	131
x11	458	445	13	5	Sem apneia	52	M	180	135
x12	528	471	57	33	Grave	40	M	179	96
x13	507	215	292	19	Moderada	57	M	171	97
x14	491	52	439	80	Grave	38	M	180	120
x15	499	299	200	16	Leve	63	M	179	104
x16	516	451	65	24	Moderada	53	M	176	85
x17	401	400	1	0	Sem apneia	27	F	158	53
x18	460	458	2	0	Sem apneia	27	M	184	72
x19	488	81	407	56	Grave	54	M	168	80
x20	514	250	264	43	Grave	51	M	179	88
x21	511	391	120	19	Moderada	53	M	176	85
x22	483	481	2	0	Sem apneia	27	F	158	53
x23	528	409	119	14	Leve	43	M	177	80
x24	430	429	1	0	Sem apneia	31	M	170	66
x25	511	220	291	48	Grave	55	M	178	90
x26	521	177	344	15	Leve	57	M	171	97
x27	499	11	488	75	Grave	60	M	176	113
x28	496	62	434	75	Grave	60	M	176	113
x29	471	471	0	0	Sem apneia	41	F	180	65
x30	512	186	326	41	Grave	44	M	177	105
x31	558	42	516	94	Grave	29	F	183	100
x32	539	114	425	72	Grave	29	F	183	100
x33	474	471	3	0	Sem apneia	28	F	169	57
x34	476	472	4	0	Sem apneia	30	F	168	56
x35	484	484	0	0	Sem apneia	31	M	184	74
<b>Média</b>	<b>494</b>	<b>307</b>	<b>187</b>	<b>27</b>	<b>-</b>	<b>44</b>	<b>-</b>	<b>175</b>	<b>86</b>

A FIGURA 39 mostra um ECG processado de um paciente do grupo controle no momento de uma respiração normal, isto é, com rótulo daquele minuto diferente de "A". De forma análoga, a FIGURA 40 exhibe o caso de um ECG de um paciente do grupo A no momento da apneia.

FIGURA 39 – ECG DE PACIENTE DO GRUPO C NO MOMENTO DE RESPIRAÇÕES NORMAIS



FONTE: Próprio autor

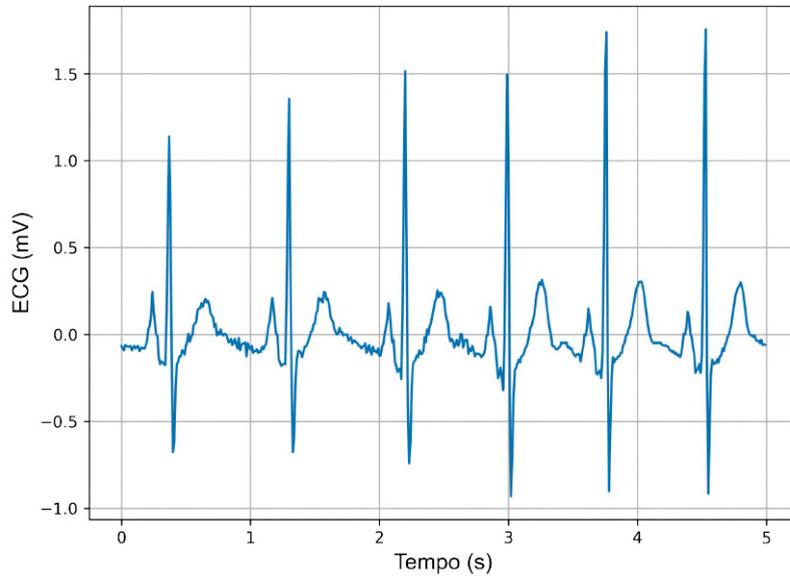
LEGENDA: ECG de paciente do grupo C no momento de respirações sem o rótulo de apneia.

A próxima seção discute o fluxo de execução do projeto, abordando uma visão geral as etapas de processamento e de desenvolvimento.

## 5.2 FLUXO DE EXECUÇÃO

O macro de fluxo de execução é mostrado nas FIGURAS 41 e 42, separado para os modelos de ML e DL respectivamente. Estas metodologias de processamento foram baseadas nos trabalhos de (BAHRAMI; FOROUZANFAR, 2022; LI et al., 2018a; CHAZAL et al., 2003; FENG et al., 2021). A leitura e processamento dos dados da Physionet são os mesmos para os modelos ML e DL. Os dados foram lidos, processados, segmentados em janelas de 1 minuto e filtrados. Em seguida foram extraídos as séries R-R de cada janela e associada ao respectivo rótulo. Então, o fluxo é diferente para cada tipo de modelo. Na FIGURA 41, dos modelos de ML, a partir de cada janela de série R-R foram calculadas características, então aplicou-se uma redução de dimensionalidade, separou-se os dados em treino, validação e teste e iniciou-se o processo de busca e seleção de hiperparâmetros, seguido de treinamento, teste e

FIGURA 40 – ECG DE PACIENTE DO GRUPO A NO MOMENTO DE APNEIA

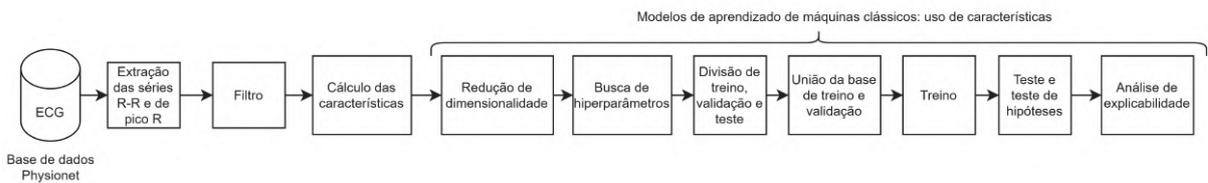


FONTE: Próprio autor

LEGENDA: ECG de paciente do grupo A no momento de respirações com o rótulo de apneia.

análise de explicabilidade do modelo. Para os modelos DL, da FIGURA 42, a diferença é que foram treinados com as séries R-R e a série de picos R, não passando pelas etapas de cálculo das características e redução de dimensionalidade, seguindo para uma etapa de interpolação, para reduzir o efeito de falsos picos (ALMUTAIRI et al., 2021), para busca e seleção de hiperparâmetros, treinamento e teste.

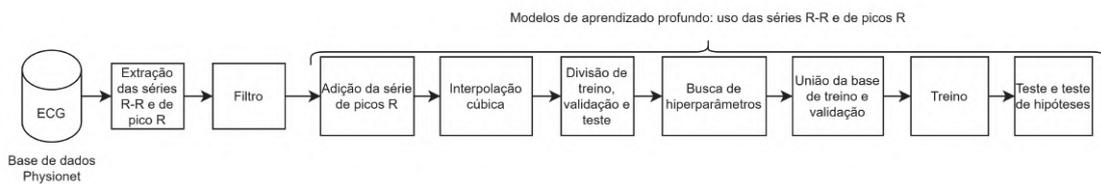
FIGURA 41 – MACRO FLUXO DE PROCESSAMENTO PARA MODELOS DE ML



FONTE: Próprio autor

LEGENDA: Macro fluxo de processamento de dados, treinamento e teste dos modelos ML.

FIGURA 42 – MACRO FLUXO DE PROCESSAMENTO PARA MODELOS DE DL



FONTE: Próprio autor

LEGENDA: Macro fluxo de processamento de dados, treinamento e teste dos modelos DL.

As próximas seções detalham cada uma das etapas de processamento.

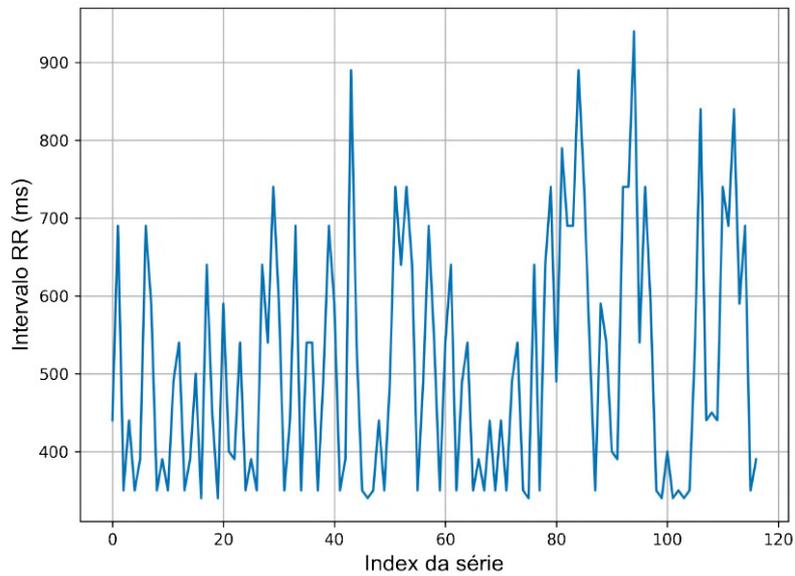
### 5.3 PRÉ-PROCESSAMENTO E FILTRAGEM DOS ECG

Inicialmente, os dados foram segmentados em janelas de 1 minuto de duração, seguindo o tamanho dos rótulos definidos na base de dados. Então, foram extraídas as séries R-R desta janela utilizando o método de Hamilton, que, de forma simplificada, consiste em aplicar diferentes filtros para detectar o complexo QRS e então são aplicadas uma série de regras para classificar os batimentos cardíacos e os picos, de modo que, os picos da onda R são identificados e a série de intervalos extraída. Todo o processo de detecção e extração é discutido em Hamilton (2002). Para implementar este algoritmo, foi utilizado o pacote de software *ECG Detectors*, desenvolvido e disponibilizado por Porr et al. (s.d.). Para ilustrar esta etapa, as FIGURAS 43 e 44 mostram as séries R-R extraídas com ECG de um paciente do grupo C no momento de respirações classificadas como sem apneia e de um paciente do grupo A no momento de apneia, respectivamente. Os pacientes são os mesmos nas FIGURAS 39 e 40, com a diferença que naquelas FIGURAS o eixo x corresponde ao tempo, tendo um total de 5 segundos, enquanto nas imagens da série R-R, o eixo x é atemporal, uma vez que cada índice da série registra o intervalo (em ms) do pico R anterior para o pico R atual. Os dados do ECG foram processados com auxílio dos pacotes *ECG detectors* (PORR et al., s.d.), *daveform database software package* (XIE et al., s.d.) e *heart rate analysis toolkit* da linguagem *python* (GENT, s.d.).

Todas as janelas de 1 minuto com as séries R-R extraídas foram então filtradas por um filtro do tipo passa média proposto por Chen et al. (2014). O filtro foi projetado com objetivo de melhorar a qualidade do sinal do ECG, que pode conter dados inesperados e que não podem ser fisiologicamente interpretados. Em geral, filtros passa média aplicam correções ao intervalo R-R com base em janelas locais e, por isso, são altamente influenciados pela mediana local e tem um desempenho pior em sucessivos intervalos anormais. O filtro proposto é capaz de corrigir sucessivos intervalos R-R (CHEN et al., 2014). O filtro funciona da seguinte forma:

- Definição de parâmetros e variáveis: neste filtro, são utilizados os parâmetros:
  - tamanho da janela do filtro, isto é, amplitude do sinal em que o filtro é aplicado. Neste trabalho, o tamanho da janela de filtro foi configurado como 10 índices da série R-R, ou 10 pontos de dados da série.
  - A variável  $\omega$  é a metade do tamanho da janela de filtro, isto é, 5 índices da série R-R.

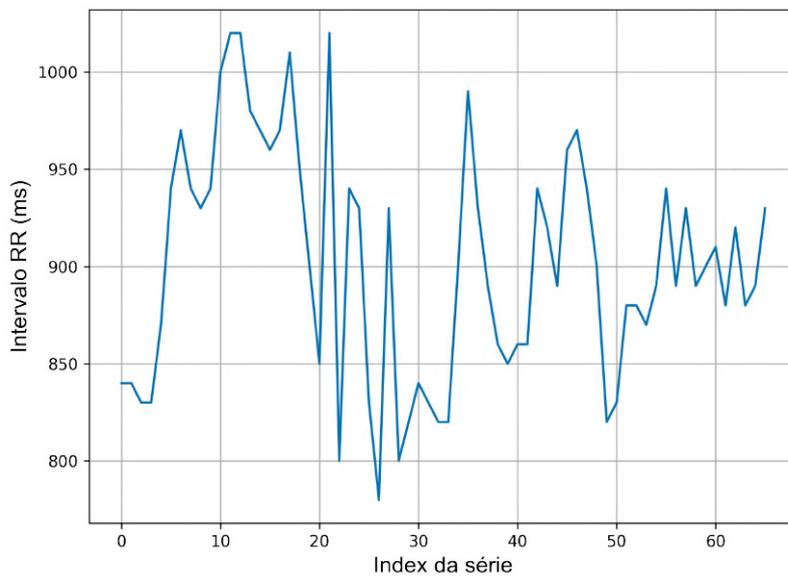
FIGURA 43 – SÉRIE R-R DE PACIENTE DO GRUPO C NO MOMENTO DE RESPIRAÇÕES NORMAIS



FONTE: Próprio autor

LEGENDA: ECG de paciente do grupo C no momento de respirações sem o rótulo de apneia.

FIGURA 44 – ECG DE PACIENTE DO GRUPO A NO MOMENTO DE APNEIA



FONTE: Próprio autor

LEGENDA: ECG de paciente do grupo A no momento de respirações com o rótulo de apneia.

- limiar inferior da série R-R, aspecto fisiológico da série R-R, chamado de LB (do inglês *Lower Bound*). Neste trabalho LB foi definido como 0,4 mV (CHEN

et al., 2014).

– limiar superior da série R-R, aspecto fisiológico da série R-R, chamado de UB(do inglês *Upper Bound*). Neste trabalho UB foi definido como 2,0 mV (CHEN et al., 2014).

- Passo 1: é determinado  $s_{min}$  como menor valor dentro da janela de 1 minuto da série RR.
- Passo 2: se  $s_{min} \geq LB$ , vai para o passo 4, caso contrário para o passo 3.
- Passo 3: é criado um vetor de dados com centro em  $s_{min}$ , que contém  $2 * \omega + 1$  pontos. Este vetor é denominado  $s_{\omega}$ , e contém os pontos:

$s_{\omega} = [s_{min-\omega}, s_{min-\omega+1}, \dots, s_{min}, s_{min+1}, \dots, s_{min+\omega}]$ . A mediana  $s_{med}$  é calculada a partir dos intervalos R-R dentro deste vetor e duas distâncias,  $d_1$  e  $d_2$  são computadas:

1.  $d_1$ , a distância absoluta da soma de  $s_{min}$  com o menor de seus vizinhos ( $s_{min-1}$  e  $s_{min+1}$ ) subtraído de  $s_{med}$ , de acordo com:

$$d_1 = |s_{min} + \min(s_{min-1}, s_{min+1}) - s_{med}|. \quad (5.1)$$

2.  $d_2$ , a distância absoluta da soma de  $s_{min}$  com maior vizinho de  $s_{min}$  e subtraído de  $s_{med}$ , dado por:

$$d_2 = 2 \left| \frac{1}{2} (s_{min} + \max(s_{min-1}, s_{min+1})) - s_{med} \right|. \quad (5.2)$$

Se  $d_1 < d_2$ , é escolhido o menor vizinho de  $s_{min}$  e estes valores são transformados em um único valor. Caso contrário, o  $s_{min}$  é fusionado com seu maior vizinho através de uma média. O algoritmo volta para o passo 1.

- Passo 4: é selecionado o maior valor da janela da série R-R,  $s_{max}$ . Se  $s_{max} \leq ECG$ , o algoritmo do filtro é interrompido. Caso contrário, o algoritmo vai para o passo 5.
- Passo 5: é selecionado um vetor de dados centralizado em  $s_{max}$  que contenha  $2 * \omega + 1$  pontos. A mediana  $s_{med}$  deste vetor é calculado. De forma semelhante, duas distâncias,  $d_3$  e  $d_4$ , são calculadas:

1.  $d_3$ , a distância absoluta entre  $s_{med}$  e  $\frac{s_{max}}{k}$ , dado por:

$$d_3 = k \left| \frac{s_{max}}{k} - s_{med} \right|. \quad (5.3)$$

onde  $k = \max(\lfloor \frac{s_{max}}{s_{med}} \rfloor)$  e o operador  $\lfloor x \rfloor$  retorna o maior inteiro não maior que  $x$ .

2.  $d_4$ , a distância absoluta entre  $s_{med}$  e a média de  $s_{max}$  e o vizinho de menor valor, dado por:

$$d_4 = 2 \left| \frac{1}{2} (s_{max} + \min[s_{max-1}, s_{max+1}]) - s_{med} \right|. \quad (5.4)$$

Se  $d_3 < d_4$ , o valor de  $s_{max}$  é dividido por  $k$ . Caso contrário os valores  $s_{max}$  e seu menor vizinho são fusionados através de uma média. O algoritmo então retoma ao passo 4.

A TABELA 12 resume as configurações utilizadas no filtro.

TABELA 12 – PARÂMETROS UTILIZADOS NO FILTRO

Parâmetro	Descrição	Valor
$2\omega$	Tamanho da janela de dados	10
LB	Lower bound em mV	0,4
UB	Upper bound em mV	2,0

FONTE: Próprio autor

Após este processo, cada sinal de série R-R obteve aproximadamente 60 pontos de dados, variando de acordo com a quantidade de picos R presente naquele segmento. Com as séries R-R, a próxima etapa para os modelos do tipo ML é a extração das características que serão utilizadas para que os modelos procurem padrões para classificação da AOS. A próxima seção aborda quais e como características foram calculadas e utilizadas.

Para os modelos do tipo DL, além das séries R-R foi extraído a série de picos R, que corresponde ao intervalo, em segundos, de cada pico R no ECG. Ambas as séries passaram por uma interpolação cúbica de 3 Hz (conforme proposto em Bahrami e Forouzanfar (2022)), ambas as séries foram segmentadas em janelas de 60 segundos e utilizadas como entrada nos modelos de DL. Após a interpolação cúbica, os sinais de entrada para os modelos de DL tiveram tamanho de 180x2 pontos de dados. Mais detalhes da otimização destes modelos são discutidos na seção 5.7.2.

## 5.4 EXTRAÇÃO DAS CARACTERÍSTICAS

As características calculadas a partir das séries R-R serão a fonte de informação para os modelos de classificação. Embora outras características podem ser utilizadas com esta base de dados, como peso, altura e sexo, para este projeto foram escolhidas apenas características do ECG, por diferentes motivos. O primeiro motivo é o objetivo é classificar AOS dentro de segmentos do ECG, ou seja, ao invés de buscar avaliar o paciente como todo, a classificação dentro de eventos do ECG. Além

disto, a base de dados é composta por uma maioria de pacientes do sexo masculino, sendo que existe uma predominância maior de AOS em indivíduos do sexo masculino (SHNEERSON, 2009). Além disto, das 7 pacientes do sexo feminino que compõe a base de dados, apenas duas apresentam AOS. A inclusão desta característica poderia induzir um viés no modelo, podendo atribuir um peso muito grande a esta característica, de modo que a capacidade de generalização do modelo poderia ser comprometida. Também existem relações entre peso e a incidência de AOS (AYONARA et al., 2014) e na base utilizada a correlação entre peso e IAH foi de 0,60, outra característica que poderia acionar viés no resultado da classificação. Por fim, a altura não parece ser um fator na ocorrência de AOS (SHNEERSON, 2009), o que poderia atrapalhar o modelo com informação irrelevante ou uma correlação espúria.

Para cada janela filtrada de série R-R de 1 minuto de duração foram calculadas diferentes características que foram utilizadas no treinamento dos modelos de ML. Estas características podem ser divididas em características do domínio do tempo e da frequência. Ao todo foram calculadas 23 características (BAHRAMI; FOROUZANFAR, 2022; EUROPEAN SOCIETY OF CARDIOLOGY THE NORTH AMERICAN SOCIETY OF PACING ELECTROPHYSIOLOGY, 1996; CLIFFORD, 2002; RICHMAN; MOORMAN, 2000):

1. *Características do domínio do tempo*: são características que calculam os aspectos estatísticos da série R-R e dos batimentos cardíacos, foram utilizadas: média do intervalo R-R (média R-R) em ms; mediana dos intervalos R-R em ms (mediana R-R); diferença entre o maior e o menor intervalo R-R em ms (amplitude R-R); desvio padrão da janela inteira (SDNN R-R); desvio padrão de dois picos R-R consecutivos (SDSD R-R); número de intervalos R-R que diferem em mais de 50 ms em toda janela (NN50); porcentagem de NN50 em toda série (pNN50); número de intervalos R-R que diferem em mais de 20 ms (NN20); porcentagem de NN20 em todo intervalo (pNN20); raiz quadrada da média dos quadrados das diferenças dos intervalos R-R (RMSSD R-R); coeficiente de variação da diferença dos intervalos R-R (CVSD); coeficiente de variação dos intervalos R-R (CVNNI); frequência cardíaca máxima do intervalo em BPM (MAXHR); frequência cardíaca mínima do intervalo em BPM (MINHR); média das frequências cardíacas do intervalo em BPM (média HR) e o desvio padrão da frequência cardíaca em BPM (STDHR).
2. *Características do domínio da frequência*: a variabilidade cardíaca possui três componentes predominantes de frequência, sendo as frequências muito baixas que vão de 0,003 a 0,04 Hz, as baixas frequências que vão 0,04 Hz até 0,15 Hz e as frequências altas de 0,15 até 0,4 Hz. Foram extraídas as potências destas três componentes, indicadas como VLF, LF e HF respectivamente. Além

destas características, também foram calculadas: a razão entre as altas e baixas frequências (LF/HF); potência total (potência total); porcentagem de potência de baixa frequência na potência total (LFnu) e porcentagem de potências de alta frequência na potência total (HFnu).

Para cálculo das características, foi utilizada o pacote de código aberto *hrv-analysis* (disponível em (CHAMPSEIX, s.d.)). As janelas que apresentaram erros durante o processo de cálculo foram removidas. Desta forma, um total de 34174 séries R-R foram extraídas. As 23 características calculadas compõe a base de características ou o espaço de características e transformam o problema em uma problema de alta dimensionalidade. A fim de estudar os efeitos da dimensionalidade nos classificadores, foram conduzidos experimentos com e sem uso de técnicas de redução de dimensionalidade. A próxima seção aborda como estas técnicas foram empregadas.

## 5.5 REDUÇÃO DA DIMENSIONALIDADE

Para a redução de dimensionalidade foram empregadas as seguintes técnicas: PCA, t-SNE e UMAP. Estas técnicas também possuem hiperparâmetros a serem sintonizados, que impactam no resultado obtido. A configuração destes hiperparâmetros foi diferente para cada uma das técnicas. Para implementação foi métodos, foi utilizado o pacote de código aberto *scikit-learn* (PEDREGOSA et al., 2011).

Para o PCA, o principal hiperparâmetro é a quantidade de componentes a serem utilizadas. O método por padrão retorna um total de componentes iguais ao número de características da base (ver seção 3.4.1), no caso 23. Para diminuir a quantidade de componentes utilizadas, e por consequência a dimensão, o PCA foi aplicado somente na base de treino, os resultados foram analisados e foi computado quanto de variabilidade cada componente explicava até se obter 99% de explicabilidade. Então o PCA foi recalculado com este limite de componentes, sendo utilizado este algoritmo para reduzir a dimensão das bases de validação e teste. O PCA foi implementado utilizando o pacote *scikit-learn* (PEDREGOSA et al., 2011).

Os hiperparâmetros do PCA foram sintonizados utilizando somente a base de treinamento. Após o processo de sintonia, este algoritmo obtido foi então aplicado nas bases de validação e teste.

O t-SNE e o UMAP possuem mais hiperparâmetros para serem ajustados. Com isto foi realizado dois experimentos com auxílio do Optuna, um para cada algoritmo. Os experimentos consistiram em aplicar a técnica de redução de dimensão em 100 iterações do Optuna, utilizando a técnica de amostragem TPE, na base de treino, em que cada iteração os hiperparâmetros variaram seguindo a busca bayesiana do Optuna e dentro de intervalos pré-definidos. Uma vez treinado, a técnica era aplicada

na base de validação e o resultado era avaliado em termos do coeficiente de silhueta (ou *silhouette score*). Como as técnicas reduzem as dimensões para 2 ou 3 e, de certa forma, agrupam o conjunto de dados em alta dimensão. Desta forma, um agrupamento eficiente dos dados em 2 ou 3 dimensões seria aquele que permitisse a visualização dos grupos de ocorrência ou não-ocorrência de AOS de forma distinta, em outras palavras, que os grupos estejam separados. Para isto, o coeficiente de silhueta utiliza dos rótulos originais da classe e compara com os grupos criados após a transformação da base para determinar o quanto uma amostra é similar as outras amostras daquele mesmo grupo (GÉRON, 2019; ROUSSEEUW, 1987). O coeficiente de silhueta,  $s(i)$  definido na equação 5.5, varia de -1 a 1, de modo que quanto maior o valor, melhor ajustado a amostra (PEDREGOSA et al., 2011; ROUSSEEUW, 1987).

$$s(i) = \frac{b(i) - a(i)}{\max[a(i), b(i)]} \quad (5.5)$$

Na equação  $a(i)$  é a distância média da amostra  $i$  para o grupo interno e  $b(i)$  é a distância média desta amostra para o grupo externo mais próximo (PEDREGOSA et al., 2011; ROUSSEEUW, 1987). Desta maneira, a busca de hiperparâmetros pelo Optuna buscou maximizar o coeficiente de silhueta.

Para o t-SNE, o número de componentes variou entre 2 e 3, o intervalo do valor inteiro do hiperparâmetro de perplexidade foi configurado para ir de 5 até 50, este hiperparâmetro tende a controlar a densidade de pontos nos grupos, quanto maior o valor, mais denso os grupos de dados tendem a ser, a taxa de aprendizado variou em números inteiros de 100 até 1000 e o hiperparâmetro exageração inicial (ou *early exaggeration*), que ajusta o quão agrupado estarão os dados nos grupos e a distância entre os grupos, variou de números decimais positivos de 1,0 até 10,0 com um passo de 0,5. O algoritmo do t-SNE do pacote *scikit-learn* foi utilizado (PEDREGOSA et al., 2011).

Para o UMAP, o número de componentes variou entre 2 e 10, a quantidade de vizinhos variou entre 2 e 100. Este hiperparâmetro controla a quantidade de pontos amostrais da vizinhança local utilizadas para aproximação. A taxa de aprendizado variou entre 0.10 e 1.00 com passos de 0.05. O UMAP implementado foi utilizando o software de código aberto disponibilizado em McInnes (s.d.). Em ambos os algoritmos a semente para geração de números aleatórios foi a mesma.

A TABELA 13 resume este processo, mostrando os intervalos utilizados na sintonia de cada hiperparâmetro.

Somente após a sintonia dos hiperparâmetros, é que os algoritmos obtidos para o t-SNE e UMAP foram aplicados na base de teste.

TABELA 13 – HIPERPARÂMETROS DAS TÉCNICAS DE REDUÇÃO DE DIMENSIONALIDADE

Técnica	Parâmetro	Intervalo
t-SNE	N° componentes	2 ou 3
	Perplexidade	de 5 até 50
	<i>Early exaggeration</i>	de 1,0 até 10,0, com passo de 0,5
	Taxa de aprendizado	de 100 até 1000
UMAP	N° componentes	de 2 a 10
	N° vizinhos	de 2 a 100
	Taxa de aprendizado	de 1,0 até 11,0 com passo de 0,05

FONTE: Próprio autor

A próxima seção discute como os dados foram divididos nas bases de treino, validação (utilizadas aqui) e teste.

## 5.6 DIVISÃO DOS DADOS

Os dados foram divididos em três grupos: treino, validação e teste, seguindo a proposta da PhysioNet, isto é, mantendo o grupo X para teste. Primeiro, foi selecionado uma amostra contendo 80% dos dados dos grupos A, B, C para treinamento, os 20% restantes foram para base de validação. A TABELA 14 resume esta divisão.

TABELA 14 – Separação das amostras em treino, validação e teste

Base	Grupos	Percentual dos Grupos	Total de Dados
Treino	A, B e C	80%	13587
Validação	A, B e C	20%	3397
Teste	X	100%	17191

Os dados das bases de treino e validação foram utilizados na busca e seleção dos hiperparâmetros de cada modelo. Os dados de teste foram utilizados no teste final do modelo.

## 5.7 SINTONIA DE HIPERPARÂMETROS

Nos experimentos deste trabalho, a otimização dos foi realizada através da ferramenta Optuna. Os modelos passaram por um processo de 100 iterações em busca dos melhores hiperparâmetros utilizando o F-1 score como métrica de otimização, com objetivo de se obter a maior acuracidade possível. Para busca dos hiperparâmetros, foi utilizado a técnica de amostragem baseada no estimador Parzen estruturado em árvore (TPE, do inglês *Tree-structured Parzen Estimator*). Este estimador consiste em um método de otimização bayesiana, que organiza o espaço de busca em uma árvore, mantendo densidade de probabilidade estimadas para possíveis configurações de hiperparâmetros separadas, usando estas estimativas para guiar a busca dentro da árvore (AKIBA et al., 2019).

Em cada iteração foram utilizadas as bases de treinamento e validação. Os modelos foram treinados na base de treinamento com os hiperparâmetros selecionados para aquela interação e têm sua acuracidade medida na base de validação. A próxima seção detalha os espaços de busca utilizados.

Os resultados de cada iteração foram salvos, bem como os valores dos hiperparâmetros, tempo total gasto durante a otimização e impacto no tempo de cada hiperparâmetro. Ao final do processo, foram selecionados os hiperparâmetros da iteração que apresentou maior acuracidade. As subseções seguintes apresentam quais hiperparâmetros de cada modelo foram otimizados. As configurações do processo de otimização são resumidas na TABELA 15.

TABELA 15 – CONFIGURAÇÃO DO PROCESSO DE OTIMIZAÇÃO E SINTONIA DE HIPERPARÂMETROS

Configuração	Valor
Ferramenta de busca	Optuna
Técnica de amostragem	TPE
Total de iterações de busca	100
Função objetivo (métrica de otimização)	Acuracidade
Direção da otimização	Maximização

FONTE: Próprio autor

### 5.7.1 Modelos de aprendizado de máquinas

Foram analisados 11 modelos de ML: DT, RF, SVM, classificador bayesiano (Nayves-Bayes), MLP, LR, aglomeração de classificadores (*Bagging*), LightGBM, XGBoost, AdaBoost, VM. Com a exceção dos modelos XGBoost e LightGBM, todos os demais foram implementados com pacote *scikit-learn* (PEDREGOSA et al., 2011). O XGBoost foi implementado com pacote de código aberto *dmlc XGBoost*, disponibilizado em Anônimo (s.d.[c]). O LightGBM foi implementado utilizando também outro pacote de código aberto disponibilizado em Anônimo (s.d.[a]).

#### 5.7.1.1 *Adaptive boosting*, AdaBoost

Outro tipo de modelo do tipo *ensemble* utilizado foi o AdaBoost. Neste modelo, os hiperparâmetros foram o algoritmo empregado, cujos valores possíveis eram "SAMME" e "SAMME.R", que são estimadores utilizados para classificação, sendo o "SAMME.R" a versão com regularização do algoritmo SAMME (GÉRON, 2019; HASTIE et al., 2009a). A taxa de aprendizado foi configurada com valores entre 0,1 e 2, com incrementações de 0,1 e o número de estimadores foi escolhido entre 1 e 1000. A TABELA 16 mostra estes intervalos.

TABELA 16 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DO ADABOOST

Hiperparâmetro	Intervalo
Algoritmo	SAMME e SAMME.R
Nº estimadores	de 1 até 1000
Taxa de aprendizado	de 0,1 até 2,0, em passo de 0,1

FONTE: Próprio autor

### 5.7.1.2 Rede Neural do tipo Multicamadas de Perceptrons

No modelo MLP o espaço de busca consistiu no total de camadas ocultas de 1 a 4, com a quantidade de perceptrons variando entre 1 e 100 para cada camada, os valores da função ativação variando entre "identidade"(sem função de ativação, retornando o próprio valor), "logística", "tangente hiperbólica"(tanh) e "retificação linear"(ReLU, do inglês *Rectified Linear Unit*); o parâmetro alfa de 0,0001 até 0,001, com passos de 0,0005. Este hiperparâmetro tem como função regular a penalização para pesos de valores elevados, ajudando a diminuir a complexidade da rede. A taxa de aprendizado variou de 0,0001 até 0,1, com passos de 0,005. A TABELA 17 resume os hiperparâmetros do espaço de busca do modelo MLP. Na TABELA, existem quatro diferentes configuração de quantidade de perceptrons, isto ocorre pois cada configuração é para uma camada específica. Caso sejam utilizadas, por exemplo, três camadas em uma determinada iteração, então o hiperparâmetro "quantidade de perceptrons 4" não é utilizado.

TABELA 17 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DO MLP

Hiperparâmetro	Intervalo
Total de camadas ocultas	de 1 até 4
Quantidade de perceptrons 1	de 1 até 100
Quantidade de perceptrons 2	de 1 até 100
Quantidade de perceptrons 3	de 1 até 100
Quantidade de perceptrons 4	de 1 até 100
Função de ativação	identidade, logística, tanh e ReLU
Alfa	de 0,0001 até 0,001 com passos de 0,0005
Taxa de aprendizado	de 0,0001 até 0,1 com passo de 0,005

FONTE: Próprio autor

NOTA: Um máximo de até 4 camadas podem ser utilizados, com isto, até 4 diferentes configurações de a depender do número de estimadores

### 5.7.1.3 Agregação de Classificadores, *Bagging*

Na agregação de classificadores, os classificadores consistiram na seguinte lista no espaço de busca: "SVM", "árvore de decisão"(DT), "regressão logística"(LR) e "k-próximos vizinhos"(KNN, do inglês *k-nearest neighbours*). Os demais hiperparâmetros foram: quantidade total de classificadores, de 10 até 1000, selecionando valores em potência de 10, a razão do número máximo de amostras retiradas da base de trei-

namento, que variou de 0,2 até 0,8 e se esta subamostragem foi com reposição ou não (no caso, uma lista com verdadeiro, indicando com reposição e falso, sem reposição). A TABELA 18 resume os hiperparâmetros do modelo *bagging*. Por limitações computacionais, cada um dos classificadores escolhidos foram treinados com os valores padrões de hiperparâmetros do pacote *scikit-learn*. Os valores dos principais hiperparâmetros são mostrados na TABELA 19. Quando o parâmetro "profundidade máxima" de uma DT recebe o valor "não definido", como é o padrão mostrado na TABELA, então os nós de decisão da árvore são expandidos até que todas as folhas sejam puras ou que todas as folhas contenham um total de amostras menor que o valor definido pelo hiperparâmetro "mínimo de amostras para divisão". Os pesos no classificador KNN tem como função balancear e controlar as fronteiras de separação das classes. Quando este hiperparâmetro é definido como "uniforme", então todos pontos dentro do grupo têm o mesmo peso (PEDREGOSA et al., 2011).

TABELA 18 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DO *BAGGING*

Hiperparâmetro	Intervalo
Classificador	SVM, DT, LR ou KNN
Quantidade de classificadores	10, 100 ou 1000
Razão de amostras retiradas da base de treinamento	de 0,2 até 0,8
Subamostragem com reposição	Verdadeiro ou falso

FONTE: Próprio autor

TABELA 19 – VALORES DOS PRINCIPAIS HIPERPARÂMETROS DOS CLASSIFICADORES USADOS NO MODELO DE *BAGGING*

Hiperparâmetro	Modelo	Valor
<i>Kernel</i>	SVM	RBF
C	SVM	1
Critério de divisão	DT	Gini
Estratégia de divisão	DT	Melhor
Profundidade máxima	DT	Não definido
Mínimo de amostras para divisão	DT	2
Mínimo de amostras em uma folha	DT	1
Penalidade	LR	L2
Número de vizinhos	KNN	5
Tipo de peso para classificação	KNN	Uniforme
Métrica de distância	KNN	Euclidiana

FONTE: Próprio autor

NOTA: Para profundidade máxima não definida em uma DT, novos nós serão adicionados até que todas as folhas sejam puras ou até que todas as folhas contenham menos amostras do que o valor do hiperparâmetro mínimo de amostras para divisão. Os pesos uniformes em um classificador KNN atribuem mesmo valor de predição para todos os pontos dentro de um grupo (PEDREGOSA et al., 2011).

#### 5.7.1.4 Árvores de Decisão

Na modelagem por DT, os hiperparâmetros do espaço de busca foram o critério para medir a qualidade da divisão dos nós, cujos valores poderiam ser índice de Gini e entropia, a estratégia para divisão do nó, se "melhor" ou "aleatória", em que a divisão do nó será de forma aleatória ou utilizará a melhor divisão possível (HASTIE et al., 2009b), a profundidade máxima da árvore, que variou de 2 até 64, seguindo potências de 2 e a quantidade mínima de amostras necessárias para dividir um nó interno, cujos valores foram escolhidos entre 2 e 10. A TABELA 20 resume estes intervalos de busca.

TABELA 20 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DE DT

Hiperparâmetro	Intervalo
Critério de divisão	Gini ou entropia
Estratégia de divisão	Melhor ou aleatória
Profundidade máxima	2, 4, 8, 16, 32 ou 64
Mínimo de amostras para divisão	de 2 até 10

FONTE: Próprio autor

#### 5.7.1.5 *Ensemble* por Voto Majoritário

Para o modelo VM, foram considerados *ensembles* de árvores de decisões (DT) e máquinas de vetores de suporte (SVM). A quantidade de estimadores foi escolhida com valores entre 2 e 4. Para os estimadores DT apenas o hiperparâmetro de profundidade máxima foi otimizado, com objetivo de simplificar a execução. Este hiperparâmetro variou de 1 a 10. No caso dos estimadores de SVM, os hiperparâmetros C e gama foram estudados. Ambos tiveram valores entre 0,00001 e 100000. Além dos estimadores, foi criado um espaço de busca para os pesos de cada modelo variando entre 0 e 1,0. A TABELA 21 mostra os hiperparâmetros e seus respectivos intervalos de busca. Na TABELA, é indicado quando o hiperparâmetro é do modelo VM, chamado de base, ou dos modelos escolhidos para serem os estimadores. Na TABELA também existem quatro valores de peso, isto acontece pois é um peso para cada estimador, logo um máximo de quatro pesos podem ser utilizados. Caso em uma determinada iteração, o número de estimadores selecionado for 2, então apenas dois pesos serão utilizados.

#### 5.7.1.6 *Light Gradient Boosting*

Para o modelo LightGBM, os hiperparâmetros são quantidade de folhas máximas para os estimadores, com valores entre 2 e 256, taxa de aprendizado entre 0,01 e 0,1; profundidade máxima para os estimadores, cujos valores foram selecionados entre

TABELA 21 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DO VM

Hiperparâmetro	Modelo	Intervalo
Tipo de modelo	VM (base)	DT ou SVM
N° estimadores	VM (base)	de 2 até 4
Profundidade máxima	DT (estimador)	de 1 até 10
C	SVM (estimador)	entre 0,00001 e 100000
Gama	SVM (estimador)	entre 0,00001 e 100000
Peso 1	VM (base)	de 0 até 1,0
Peso 2	VM (base)	de 0 até 1,0
Peso 3	VM (base)	de 0 até 1,0
Peso 4	VM (base)	de 0 até 1,0

FONTE: Próprio autor

NOTA: um máximo de até 4 pesos podem ser utilizados a depender do número de estimadores

2 e 16; quantidade mínima do somatório de pesos necessário em uma folha (as vezes referida como *child*, do inglês criança), com valores entre 1 e 100; tamanho da razão da amostragem para instância de treinamento, com valores entre 0,1 e 1; tamanho da razão de amostras de colunas quando utilizados em DT, que variou de 0,1 e 1; os reguladores de peso alfa e lambda, ambos tendo valores escolhidos entre  $1 \times 10^{-9}$  e 10 e quantidade de estimadores escolhida entre 50 e 500. A TABELA 22 apresenta estes valores para melhor visualização.

TABELA 22 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DO LIGHTGBM

Hiperparâmetro	Intervalo
Quantidade de folhas máxima	de 2 até 256
Profundidade máxima	de 2 até 16
Soma mínima de pesos	de 1 até 100
Razão de amostragem para instância de treinamento	de 0,1 até 1
Razão de amostras de coluna	de 0,1 até 1
Regulador alfa	de $1 \times 10^{-9}$ até 10
Regulador lambda	de $1 \times 10^{-9}$ até 10
Taxa de aprendizado	de 0,01 até 0,1
Número de classificadores	de 50 até 500

FONTE: Próprio autor

#### 5.7.1.7 Regressão Logística

O modelo LR possui três hiperparâmetros principais: penalidade, C e peso da penalidade L1 (quando a mesma é aplicada). Para este hiperparâmetros, os espaços de busca foram penalidade com "L1", "L2" e "*elastic net*", para C com valores entre 0,1 e 10 e para o peso da penalidade "L1" com valores 0,01 até 0,99. Quando o tipo de penalidade é diferente de "L1", ou seja "L2" ou "*elastic net*", este último hiperparâmetro não é utilizado. a TABELA 23 mostra este intervalo de espaço de busca.

TABELA 23 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DO MLP

Hiperparâmetro	Intervalo
Penalidade	L1, L2 ou <i>elastic net</i>
C	de 0,1 até 10
Peso da penalidade "L1"	de 0,01 até 0,99

FONTE: Próprio autor

NOTA: O hiperparâmetro peso da penalidade L1 é utilizando somente quando o hiperparâmetro penalidade utilizado é L1.

#### 5.7.1.8 Naive-Bayes

Para o classificador Naive-Bayes, não foi realizado ajuste de hiperparâmetros, pois o modelo calcula todas as variáveis com base apenas nos dados utilizados.

#### 5.7.1.9 Floresta Aleatória

Para o modelo RF foram avaliados o número de estimadores, 10 a 1000, o critério de medida da divisão do nó, que foi selecionado entre os valores "índice de Gini" e "entropia"; profundidade máxima da árvore, cujos valores variaram de 2 a 32, mas seguindo as potências de 2 (4, 8, 16) e a quantidade mínima de amostras necessárias antes de realizar uma divisão do nó, que variaram entre 2 e 10. Para facilitar a visualização, a TABELA 24 mostra os intervalos dos hiperparâmetros que foram ajustados.

TABELA 24 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DA RF

Parâmetro	Intervalo
Nº estimadores	de 10 a 1000
Critério divisão de nó	índice de Gini ou entropia
Profundidade máxima	2, 4, 8, 16 ou 32
Quantidade mínima de amostras	de 2 a 10

FONTE: Próprio autor

#### 5.7.1.10 Máquina de Vetores de Suporte

No modelo SVM, o *kernel* foi escolhido entre os tipos "linear", "polinomial", "RBF" (do inglês *radial basis function*) e "sigmoide", enquanto o hiperparâmetro C teve valores indo de 0,1 até 2,0, com um passo de 0,1. Estes intervalos de busca são mostrados na TABELA 25.

TABELA 25 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DA RF

Parâmetro	Intervalo
N° estimadores	de 10 a 1000
Critério divisão de nó	índice de Gini ou entropia
Profundidade máxima	2, 4, 8, 16 ou 32
Quantidade mínima de amostras	de 2 a 10

FONTE: Próprio autor

#### 5.7.1.11 *Extreme gradient boosting*, XgBoost

A TABELA 26 resume os hiperparâmetros estudados para o modelo XgBoost. A taxa de aprendizado variou de 0,1 até 2,0, em incrementos de 0,1, o número de estimadores (ou classificadores) variou entre 1 e 1000, a razão de amostragem de dados utilizada para treinamento de instâncias individuais de cada estimador variou de 0,1 até 1,0 (total da base de treinamento) e, por fim, a profundidade máxima teve valores entre 1 e 10.

TABELA 26 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DO XGBOOST

Parâmetro	Intervalo
Taxa de aprendizado	de 0,1 até 2,0 em incrementos de 0,1
N° de estimadores	de 1 até 1000
Razão de amostragem	de 0,1 até 1,0
Profundidade máxima	de 1 a 10

FONTE: Próprio autor

Uma vez terminado o processo de sintonia dos hiperparâmetros, os modelos obtidos foram testados, para se calcular o desempenho de cada um. A próxima seção detalha a metodologia destes testes.

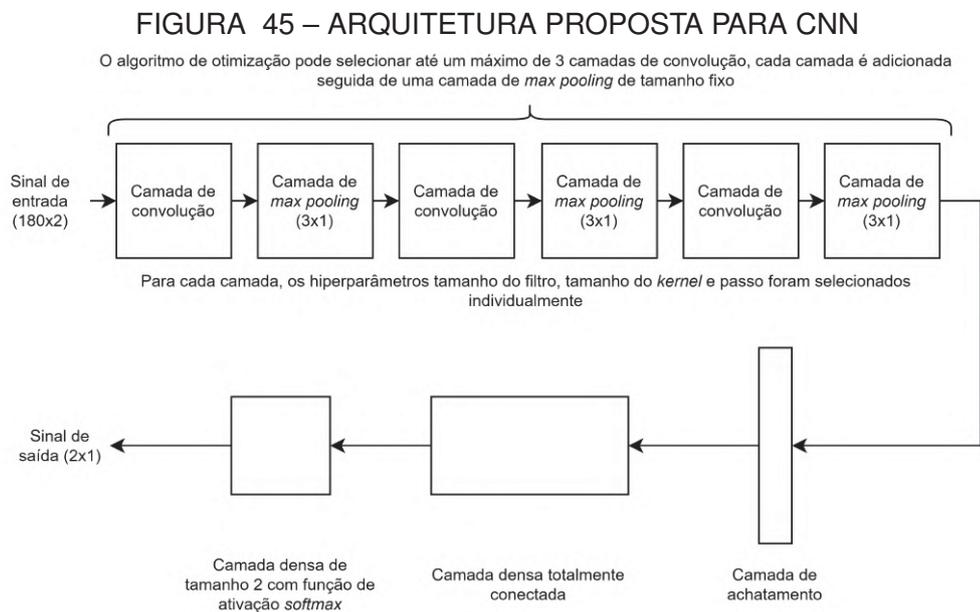
### 5.7.2 Modelos de aprendizado profundo

Esta seção detalha o processo de otimização dos hiperparâmetros dos modelos de DL estudados. Ao invés de utilizar arquiteturas conhecidas, a otimização foi configurada de tal forma que a ferramenta Optuna testasse diferentes combinações de arquiteturas, variando as camadas das redes, selecionando a melhor combinação. Os modelos de DL foram implementados utilizando as bibliotecas Keras (CHOLLET et al., 2015) e TensorFlow ((ABADI et al., 2015)).

#### 5.7.2.1 Redes neurais convolucionais, CNN

Para rede do tipo CNN, até três camadas de convolução poderiam ser adicionadas em sequências. Para cada camada, os parâmetros de busca foram: a quantidade de funções de convolução naquela camada, variando de 1 até 3, com um filtro de tamanho escolhido entre os valores 32, 64, 128 ou 256, com o tamanho *kernel* de

3, 5 ou 7, a função ativação escolhida entre linear ou ReLU e o tamanho do passo (*stride*) de 1 ou 2. Essas escolhas são individuais para cada camada de convolução. A rede é construída de tal forma que para cada camada de convolução, as funções de convolução são adicionadas em sequência, seguido de uma camada fixa de camada de max pooling de tamanho fixo 3x1. O tamanho dos dados trafegados na rede não se altera, isto é, não existe uma redução de dimensão entre as camadas. Após as camadas de convolução, é adicionado uma camada fixa de achatamento (*flatten*). Para facilitar a visualização, a FIGURA 45 apresenta como esta arquitetura de rede pode ser construída pelo algoritmo de otimização.



FONTE: Próprio autor

LEGENDA: Arquitetura de rede proposta para o algoritmo de otimização selecionar os hiperparâmetros

A saída das camadas de convolução é adicionada a camada densa, isto é, uma ANN totalmente conectada. Esta ANN teve os hiperparâmetros selecionados com total de unidades escolhidos entre 32, 64, 128, 256 e 512, com função de ativação escolhida entre linear e ReLU. Finalmente, é adicionada uma camada fixa de ANN de tamanho 2 para separação das classes. A TABELA 27 resume o espaço de busca para CNN.

### 5.7.2.2 Redes neurais recorrentes do tipo LSTM

Para o modelo LSTM, os sinais de entrada passaram por uma transformação de estrutura de modo que tivesse o tamanho 2x180, necessário para entrada nas camadas de LSTM. Para construir a rede, foram selecionadas de 1 até 3 camadas de LSTM. Para cada camada, foi adicionada uma função de LSTM com total de unidades

TABELA 27 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DAS CNN

Hiperparâmetro	Intervalo
Quantidade de camadas de convolução	de 1 até 3
Quantidade de funções de convolução por camada	de 1 até 3
Tamanho do filtro	32, 64, 128 ou 256
Tamanho do <i>kernel</i>	3, 5 ou 7
Função de ativação da camada de convolução	ReLU ou linear
Tamanho do passo ( <i>strides</i> )	1 ou 2
Total de neurônios na camada densa	32, 64, 128, 256 ou 512
Função de ativação de camada densa	ReLU e linear

FONTE: Próprio autor

que variou de 10 até 128 e uma camada de *dropout* com uma taxa escolhida entre 0,01, 0,05, 0,10, 0,15, 0,20, 0,25 ou 0,30. As camadas de LSTM foram adicionadas em sequência e a saída conectada para uma camada de achatamento (*flatten*) para então conectar em uma camada densa (ANN totalmente conectada), com tamanho escolhido entre 8, 16, 32, 64, 128, 256 e 512 e função de ativação escolhida entre linear ou ReLU. Finalmente, conectada em uma camada densa de tamanho fixo de 2 para separação das classes. A TABELA 28 mostra os intervalos de busca para cada hiperparâmetro do modelo LSTM.

TABELA 28 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DO LSTM

Hiperparâmetro	Intervalo
Quantidade de camadas de LSTM	de 1 até 3
Quantidade de unidades de LSTM	de 10 até 128
Taxa de dropout	0,01, 0,05, 0,10, 0,15, 0,20, 0,25 ou 0,30
Total de neurônios na camada densa	8, 16, 32, 64, 128, 256 ou 512
Função de ativação de camada densa	ReLU e linear

FONTE: Próprio autor

### 5.7.2.3 Redes neurais do tipo híbrida entre convolucionais e recorrentes

Para a rede híbrida, foram configuradas três camadas convolucionais, em que foram configurados o tamanho do filtro, escolhido entre tamanhos 32, 64, 128, 256, 512, o tipo de função de ativação, entre linear e ReLU e a aplicação ou não de uma camada de normalização. As três camadas foram adicionadas com a função de convolução seguida da camada de normalização (quando houve). Para manter a rede coesa, o tamanho do *kernel* e do passo (*stride*) foram mantidos fixos de 3 e 1, respectivamente. Cada camada de convolução foi conectada a uma cada *textitmax pooling* fixa de tamanho 3x1 e passo de tamanho 1, com exceção da última em que o passo teve tamanho 2x1.

Após as camadas de convolução, a saída uma transformação de tamanho baseado no valor do último filtro de convolução. Foram então adicionadas de 1 até 3

camadas de LSTM. Para cada camada de LSTM, foram configurados o número de unidades da camada, que foi escolhido entre 32, 64, 128, 256, 512. Esta configuração não foi aplicada para a primeira camada de LSTM que teve seu valor baseado no tamanho do último filtro de função de convolução, sendo um quarto deste valor. Após cada camada de LSTM foi adicionada uma camada de *dropout* com uma taxa escolhida entre 0,01, 0,05, 0,10, 0,15, 0,20, 0,25 ou 0,30.

A saída das camadas de LSTM foi conectada em uma camada de achatamento (*flatten*) para então ser conectada em uma camada densa (ANN totalmente conectada) com quantidade de unidades de tamanho escolhido entre 32, 64, 128, 256 e 512 e função de ativação escolhida entre ReLU e linear e finalmente em uma camada densa de tamanho 2 com ativação do tipo *softmax* para separação das classes.

Na TABELA 29 é mostrado o intervalo de busca dos hiperparâmetros para rede híbrida.

TABELA 29 – INTERVALO DE BUSCA DOS HIPERPARÂMETROS DA REDE HÍBRIDA

Hiperparâmetro	Intervalo
Quantidade de camadas de LSTM	de 1 até 3
Quantidade de unidades de LSTM	de 10 até 128
Taxa de dropout	0,01, 0,05, 0,10, 0,15, 0,20, 0,25 ou 0,30
Total de neurônios na camada densa	8, 16, 32, 64, 128, 256 ou 512
Função de ativação de camada densa	ReLU e linear

FONTE: Próprio autor

Uma vez finalizado a seleção de hiperparâmetros dos modelos de DL e de ML, os modelos seguiram para etapa de teste, em que os resultados foram medidos. A próxima seção detalha esta etapa.

## 5.8 AVALIAÇÃO DE DESEMPENHO E ANÁLISE DE EXPLICABILIDADE

Os modelos foram treinados novamente, desta vez utilizando a combinação da base de treinamento com a de validação, com os hiperparâmetros selecionados pelo Optuna. Em seguida os modelos foram testados com a base de teste, tendo sua acuracidade, sensibilidade, especificidade e F-1 *score* medidos, conforme descrito na subseção 3.2.1. Conforme descrito na seção 5.5, nos experimentos realizados com redução de dimensionalidade, a técnica de redução foi sintonizada utilizada a base de treino (e validação para os casos do t-SNE e UMAP), para então ser aplicada na base de validação (no caso do PCA) e teste (todos os casos).

Para o cálculo dos valores SHAP nos modelos ML foram utilizados amostras de 200 dados aleatórios extraídos da base de teste, sendo 100 amostras de eventos de AOS e 100 amostras de eventos não-AOS. O cálculo excluiu os modelos VM, SVM e

*bagging* por limitações computacionais. Os cálculos foram feitos com auxílio do pacote de código aberto disponibilizado em Lundberg (s.d.).

## 5.9 EXPERIMENTOS PROPOSTOS

Para os modelos do tipo ML foram realizados quatro tipos diferentes de experimento: sem a aplicação de nenhuma técnica de redução de dimensionalidade, isto é, utilizando todas as 23 características (ver seção 5.4) (1) e com aplicação das técnicas PCA (2), t-SNE (3) e UMAP (4), conforme descrito na seção 5.5. Além das métricas descritas na seção 5.8, foram coletados os tempos totais de cada experimento, tempo para sintonizar os hiperparâmetros de cada modelo e tempo de inferência de cada modelo. A TABELA 30 resume os quatros experimentos estudados para os modelos de ML, atribuindo números a estes experimentos, para facilitar a referência nos capítulos de resultados e discussão (6 e 7).

TABELA 30 – EXPERIMENTOS COM OS MODELOS DE ML

<b>Número do experimento</b>	<b>Técnica de redução de dimensão aplicada</b>
I	Sem redução (todas as características)
II	PCA
III	t-SNE
IV	UMAP

FONTE: Próprio autor

Para os modelos de DL, como foram utilizado os próprios sinais da série RR e amplitude dos picos R, apenas um experimento foi conduzido (5).

## 5.10 RESUMO

Neste capítulo foi apresentado a metodologia seguida para execução dos experimentos. Iniciando com o processamento de sinal do ECG, detalhando o filtro aplicado e a forma de extração dos intervalos R-R. Em seguida, foram apresentadas as características utilizadas e como foram calculadas. O processo de redução de dimensão também foi explicado, detalhando como as técnicas PCA, t-SNE e UMAP foram sintonizadas para o problema. Por fim, é detalhado o processo de otimização dos hiperparâmetros dos modelos com auxílio do Optuna, o cálculo final dos resultados e os experimentos realizados.

## **Parte III**

### **Resultados**

## 6 RESULTADOS

Este capítulo apresenta os resultados obtidos. Primeiro é mostrado os resultados obtidos na sintonia das técnicas de redução de dimensão, t-SNE e UMAP. Após isto, é apresentado os resultados dos modelos ML, primeiro de uma forma geral, em seguida, focando nos melhores resultados obtidos, para então, apresentar os hiperparâmetros selecionados e resultados de acurácia de cada modelo, de forma resumida. Para os leitores interessados, o capítulo 1 detalha o restante dos resultados de cada modelo. Por fim, são apresentados os resultados obtidos com os modelos de DL.

Os gráficos foram obtidos com uso de diferentes pacotes da linguagem *python*. De forma geral, todos os gráficos foram editados utilizando os pacotes *matplotlib* (HUNTER, s.d.) e *seaborn* (WASKOM, s.d.). Os gráficos envolvendo os resultados de otimização foram obtidos pelo próprio pacote do Optuna (AKIBA et al., 2019) e os gráficos dos resultados da explicabilidade foram obtidos com auxílio do pacote SHAP (LUNDBERG, s.d.).

### 6.1 RESULTADOS DA SINTONIA DAS TÉCNICAS DE REDUÇÃO DE DIMENSÃO

As técnicas t-SNE e UMAP tiveram seus hiperparâmetros sintonizados antes de serem utilizadas nos experimentos. A TABELA 31 mostra os resultados obtidos. Na TABELA, é possível observar que ambos os resultados tiveram um *silhouette score* baixo, isto indica que o agrupamento realizado, diminuindo as dimensões, não conseguiu separar bem as classes de pacientes. De todo o modo, estes dados em menor dimensão serão utilizados pelos modelos para realizar uma nova separação das classes. O t-SNE apresentou o maior tempo de processamento, com mais de três vezes o tempo para sintonizar o UMAP.

TABELA 31 – RESUMO DA SINTONIA DOS HIPERPARÂMETROS DAS TÉCNICAS DE REDUÇÃO DE DIMENSIONALIDADE

Técnica	<i>Silhouette Score</i>	Tempo total execução aproximado (minutos)
t-SNE	0,044	285
UMAP	0,024	79

FONTE: Elaborada pelo autor

As TABELAS 32 e 33 mostram os valores selecionados para os hiperparâmetros do t-SNE e UMAP, respectivamente. A maior dimensionalidade possível, dimensão 3, foi selecionada para o t-SNE. Este valor também foi selecionado para UMAP, de um máximo de 10 dimensões possíveis, a otimização encontrou o melhor valor com três

dimensões. Desta forma, de 23 dimensões foram reduzidas para 3. Os hiperparâmetros de perplexidade do t-SNE e o número de vizinhos do UMAP controlam, de certa forma, a densidade dos grupos e quantidade de informação local retida versus a quantidade de informação global, isto é, controlam quanto da informação local em alta dimensão vai ser retida na projeção em baixa dimensão (deixando os grupos mais ou menos denso). Em ambos os casos, valores mais elevados tendem a criar grupos mais denso e priorizam a informação local em detrimento da informação geral. Ou seja, os dados na projeção ficaram mais espaçados ou dispersos (o que contribui para os valores baixos de *silhouette score*). Isto pode ter acontecido pois, como objetivo do otimizador era determinar agrupamentos que melhor separavam as classes, as técnicas podem ter criados grupos mais distantes um dos outros.

TABELA 32 – HIPERPARÂMETROS SELECIONADOS PARA A TÉCNICA T-SNE

Hiperparâmetro	Valor
Nº de componentes	3,0
Perplexidade	9,0
Exagero inicial <i>early exaggeration, em inglês</i>	8,5
Taxa de aprendizado	431,0

FONTE: Elaborada pelo autor

TABELA 33 – HIPERPARÂMETROS SELECIONADOS PARA A TÉCNICA UMAP

Hiperparâmetro	Valor
Nº de componentes	3,0
Nº de vizinhos	42,0
Taxa de aprendizado	0,1

FONTE: Elaborada pelo autor

Para técnica PCA, foram selecionados 11 das 23 componentes. Reduzindo, assim, a dimensão para 11. Após a sintonia destes hiperparâmetros, os dados de dimensão reduzidas foram utilizadas pelos modelos para treinamento, validação e teste. A próxima seção mostra o impacto de cada uma destas técnicas nas previsões dos modelos de ML.

## 6.2 RESULTADOS DOS MODELOS DE APRENDIZADO DE MÁQUINA

Esta seção apresenta os resultados obtidos pelos modelos de ML. A primeira subseção detalha os resultados gerais dos experimentos, com os tempos de execução e métricas acuracidade obtidos. A seção seguinte detalha os resultados da explicabilidade e da otimização para os melhores modelos obtidos. As demais seções apresentam para cada um dos modelos, de forma separada, os hiperparâmetros selecionados e outras métricas coletadas.

### 6.2.1 RESULTADOS GERAIS

As TABELAS 34, 35, 36 e 37 apresentam os resultados de acuracidade, *recall*, precisão e F-1 *score* na base de teste para os experimentos sem redução de dimensão (I), com uso do PCA (II), com t-SNE (III) e com UMAP (IV). Os resultados foram obtidos com os melhores hiperparâmetros encontrados durante a otimização. As linhas destacadas em cada uma das tabelas mostra o modelo que obteve maior valor de F-1 *score*. Cada experimento foi executado uma única vez, devido a restrições computacionais. Em todos os experimentos, a SVM obteve maior F-1 *score*, com experimento 2 apresentando o maior F-1 *score* de todos. Em todos os experimentos e para todos os modelos, a acuracidade ficou abaixo de 0,800, mas este fator não impactou tanto no F-1 *score*, devido a valores mais elevados de *recall* e precisão, em quase todos os modelos. Isto demonstra que, apesar de erros eventuais na faixa de 30% das inferências, a maioria dos modelos conseguiu distinguir bem as classes, tendo uma confiança elevada nos verdadeiros negativos classificados.

TABELA 34 – RESULTADOS DAS MÉTRICAS DE DESEMPENHO OBTIDAS PARA OS MODELOS DE CLASSIFICAÇÃO SEM REDUÇÃO DE DIMENSIONALIDADE

Modelo	Acuracidade	Recall	Precisão	F1-score
XGBoost	0,753	0,794	0,813	0,804
RF	0,763	0,802	0,821	0,811
<b>SVM</b>	<b>0,770</b>	<b>0,800</b>	<b>0,839</b>	<b>0,819</b>
Naive-Bayes	0,652	0,822	0,560	0,666
AdaBoost	0,772	0,810	0,827	0,818
VM	0,757	0,780	0,846	0,812
ANN	0,743	0,755	0,866	0,807
LR	0,744	0,763	0,854	0,806
LightGBM	0,757	0,796	0,818	0,807
Bagging	0,760	0,799	0,820	0,809
DT	0,733	0,786	0,782	0,784

FONTE: Elaborada pelo autor

Os tempos totais de execução de cada experimento são mostrados na TABELA 38, onde os experimentos foram executados nas configurações de máquinas descritas no capítulo 5. Na TABELA, os tempos correspondem ao tempo de otimização dos hiperparâmetros, não estando computados os tempos de processamento dos dados, inferência dos modelos e redução da dimensionalidade. Para os experimentos de II até IV, também não está computado o tempo para sintonia dos hiperparâmetros das técnicas de redução de dimensionalidade (mostrados na TABELA 31), estes hi-

TABELA 35 – RESULTADOS DAS MÉTRICAS DE DESEMPENHO OBTIDAS PARA OS MODELOS DE CLASSIFICAÇÃO COM APLICAÇÃO DO PCA

Modelo	Acuracidade	Recall	Precisão	F1-score
XGBoost	0,738	0,777	0,809	0,793
RF	0,752	0,786	0,825	0,805
<b>SVM</b>	<b>0,772</b>	<b>0,800</b>	<b>0,843</b>	<b>0,821</b>
Naive-Bayes	0,701	0,755	0,768	0,762
AdaBoost	0,763	0,791	0,841	0,815
VM	0,737	0,758	0,847	0,800
ANN	0,744	0,777	0,825	0,800
LR	0,739	0,757	0,854	0,803
LightGBM	0,747	0,784	0,817	0,800
Bagging	0,748	0,786	0,818	0,801
DT	0,718	0,775	0,769	0,772

FONTE: Elaborada pelo autor

TABELA 36 – RESULTADOS DAS MÉTRICAS DE DESEMPENHO OBTIDAS PARA OS MODELOS DE CLASSIFICAÇÃO COM APLICAÇÃO DO T-SNE

Modelo	Acuracidade	Recall	Precisão	F1-score
XGBoost	0,725	0,764	0,806	0,785
RF	0,730	0,771	0,803	0,787
<b>SVM</b>	<b>0,742</b>	<b>0,762</b>	<b>0,849</b>	<b>0,803</b>
Naive-Bayes	0,729	0,752	0,839	0,793
AdaBoost	0,738	0,777	0,808	0,793
VM	0,741	0,765	0,841	0,801
ANN	0,701	0,743	0,792	0,766
LR	0,701	0,738	0,803	0,769
LightGBM	0,736	0,773	0,813	0,792
Bagging	0,733	0,774	0,806	0,789
DT	0,704	0,760	0,765	0,762

FONTE: Elaborada pelo autor

TABELA 37 – RESULTADOS DAS MÉTRICAS DE DESEMPENHO OBTIDAS PARA OS MODELOS DE CLASSIFICAÇÃO COM APLICAÇÃO DO UMAP

Modelo	Acuracidade	Recall	Precisão	F1-score
XGBoost	0,749	0,777	0,834	0,805
RF	0,727	0,766	0,806	0,785
<b>SVM</b>	<b>0,750</b>	<b>0,759</b>	<b>0,875</b>	<b>0,813</b>
Naive-Bayes	0,745	0,767	0,846	0,804
AdaBoost	0,741	0,771	0,829	0,799
VM	0,738	0,740	0,892	0,809
ANN	0,746	0,754	0,877	0,811
LR	0,732	0,764	0,821	0,792
LightGBM	0,742	0,776	0,822	0,798
Bagging	0,731	0,772	0,804	0,788
DT	0,737	0,746	0,872	0,804

FONTE: Elaborada pelo autor

parâmetros foram sintonizados previamente. Quanto maior a dimensão do espaço de características, maior é o tempo para treinar os modelos, portanto, diminuir as dimensões reduz o tempo de treinamento e, por consequência, o tempo de otimização de sintonia dos hiperparâmetros. Isto é observado na TABELA, com a exceção do experimento III, em que a transformação para três dimensões aplicada com uso do

t-SNE teve um tempo de sintonia maior que do experimento II, com a utilização de 11 componentes. O uso das técnicas de redução de dimensionalidade diminuiu, na média, em quase 280 minutos ou cerca de 35% o tempo de otimização.

TABELA 38 – TEMPO TOTAL DE EXECUÇÃO DOS EXPERIMENTOS

<b>Experimento</b>	<b>Tempo total de execução (min)</b>
I - Sem redução	778
II - PCA	492
III - t-SNE	565
IV - UMAP	452

FONTE: Elaborada pelo autor

A próxima subseção mostra os resultados da explicabilidade dos melhores resultados.

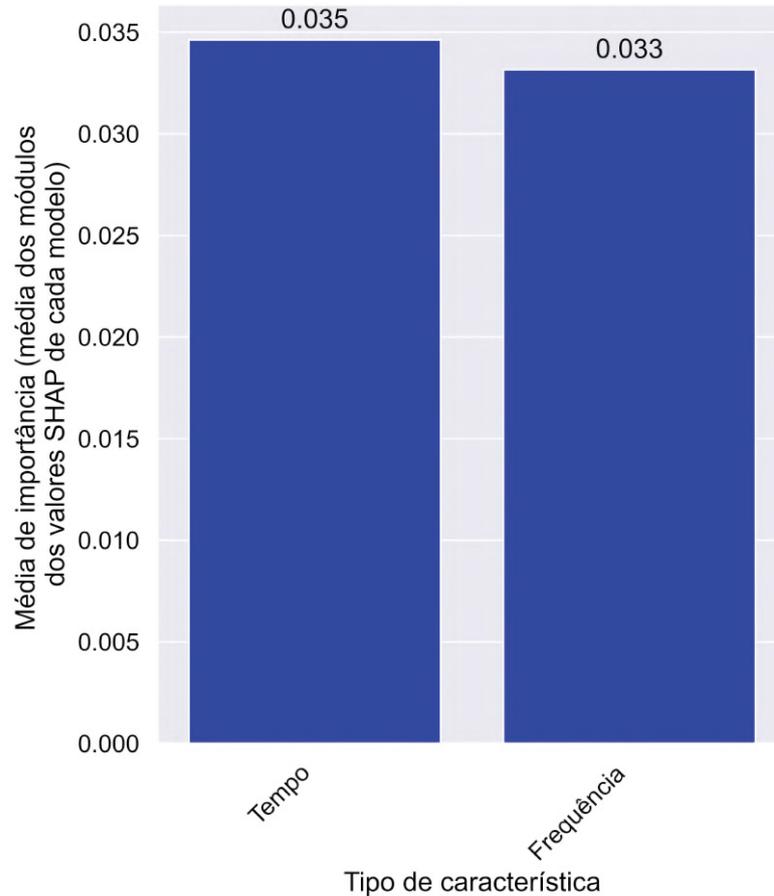
### 6.2.2 Resultados de explicabilidade

Nesta seção são apresentados os resultados de explicabilidade, primeiro analisando de uma forma geral o experimento I, em que as características utilizadas tem relação direta com os sinais fisiológicos dos pacientes. Os resultados gerais dos experimentos II a IV são apresentados no anexo (1). Na seção seguinte, é apresentado os resultados para um modelo específico. Como o melhor resultado obtido foi pela SVM no experimento I, mas por razões de limite computacionais, não foram calculados valores SHAP para SVM, o foco da seção será o AdaBoost, segundo colocado nos resultados da TABELA 34. Novamente, todos os resultados de explicabilidade para todos os modelos e experimentos estão disponíveis no anexo.

#### 6.2.2.1 Resultados gerais de explicabilidade

No experimento I, existem 23 características divididas em dois grupos, temporais e do domínio da frequência. A FIGURA 46 mostra a média de valores SHAP de cada um dos domínios. Este cálculo foi feito calculando-se a média de cada valor SHAP calculado na amostra de análise de 200 dados para cada modelo, para então, agregar estes resultados por domínio, para então, calcular a média destes domínios com todos os modelos do experimento. Na FIGURA, a média de importância de cada domínio ficou próxima, com uma diferença de 0,02. De forma geral, tanto o domínio da frequência como o domínio do tempo tiveram impactos parecidos nas decisões dos models. Vale ressaltar que, enquanto do domínio temporal contava com 16 características, o domínio da frequência contava com apenas 7. Como a média dos impactos ficou próxima, isoladamente as características do domínio da frequência desempenharam um papel maior no resultado de inferência do modelo. Isto ficará mais evidente ao analisar os demais resultados.

FIGURA 46 – IMPORTÂNCIA GERAL DE CADA DOMÍNIO DE CARACTERÍSTICAS PARA O EXPERIMENTO I

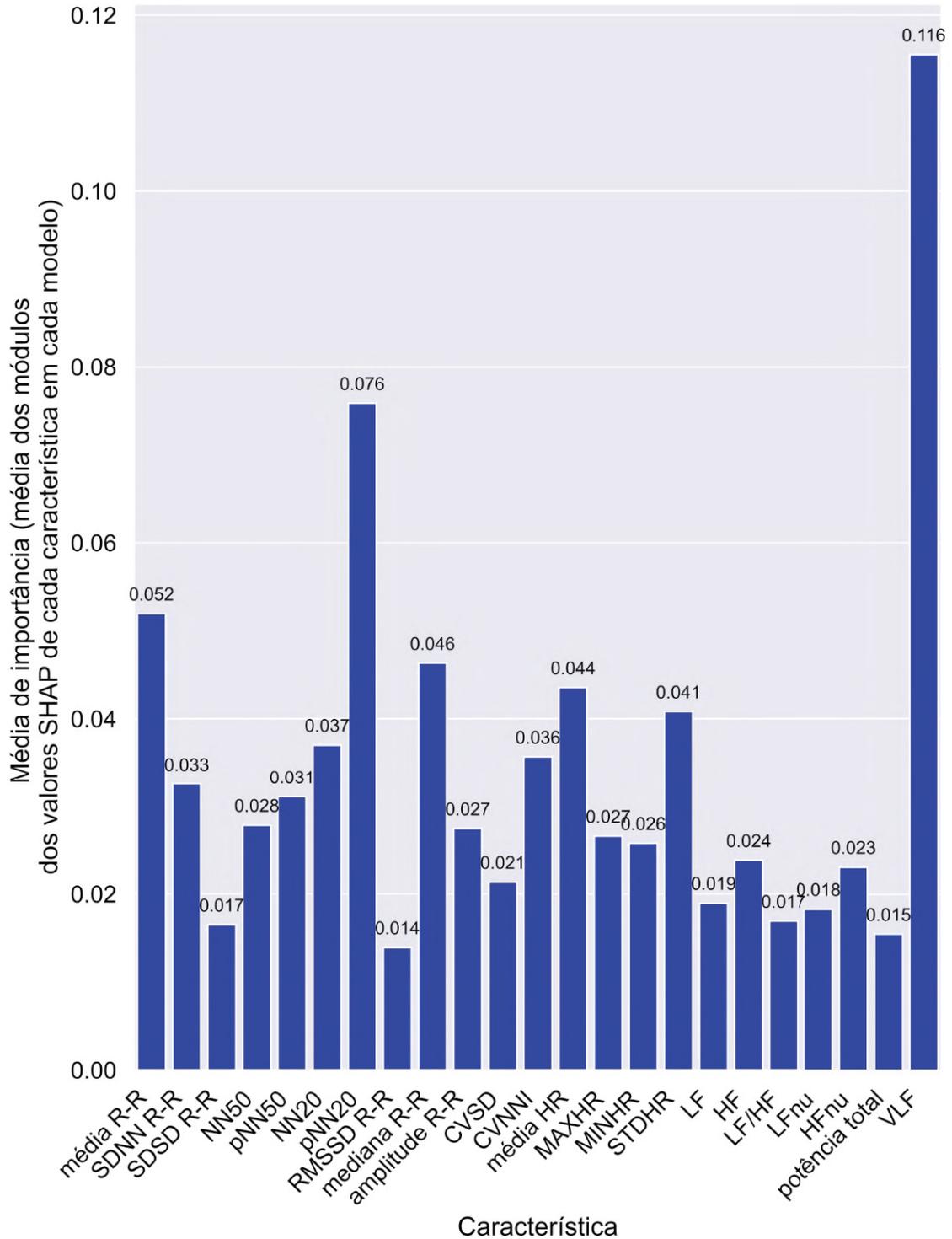


FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada domínio características.

A FIGURA 47 apresenta um resultado semelhante ao discutido na FIGURA 46, mas desta vez expandindo nas 23 características. A característica que, na média, mais impactou as decisões dos modelos foi a VLF, do domínio da frequência, seguida pela pNN20 do domínio do tempo, o que significa dizer que, das 200 amostras analisadas, estas características que mais contribuíram para explicar os resultados. Dentre as características que menos ajudaram a explicar as inferências dos modelos, estão RMSSD, potência total, pNN50 e LF/HF. A análise da FIGURA 47 auxilia na compreensão do problema, uma vez que cada uma destas características corresponde a fatores dos sinais fisiológicos. Também pode levantar novas hipóteses a serem testadas, como o efeito da eliminação das características de menor média de importância. Esta análise deve ser complementada com as análises individuais de cada modelo. A próxima seção aborda um caso de estudo para o modelo AdaBoost.

FIGURA 47 – IMPORTÂNCIA GERAL DE CADA DE CARACTERÍSTICAS PARA O EXPERIMENTO I



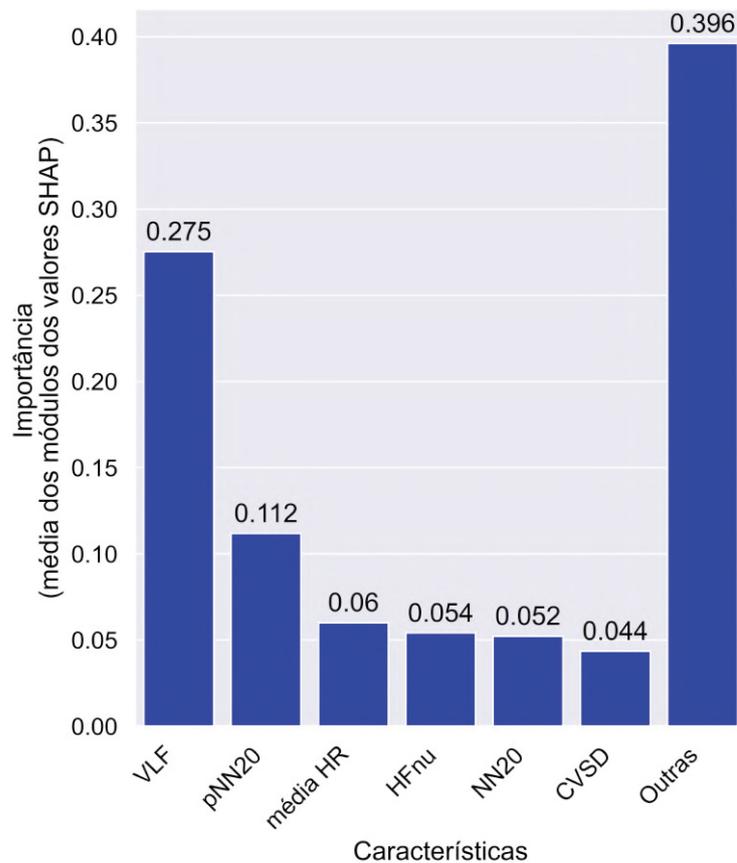
FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para cada modelo no experimento I.

### 6.2.2.2 Estudo de caso de explicabilidade: *adaptive boosting*

O AdaBoost obteve o segundo melhor desempenho no experimento I. A FIGURA 48 mostra a importância média de cada característica. A característica VLU e pNN20, assim como observado na FIGURA 47, foram as características que melhor explicaram o resultado obtido, com VLU obtendo a maior média dos valores SHAP.

FIGURA 48 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO ADABOOST NO EXPERIMENTO I



FONTE: Próprio autor

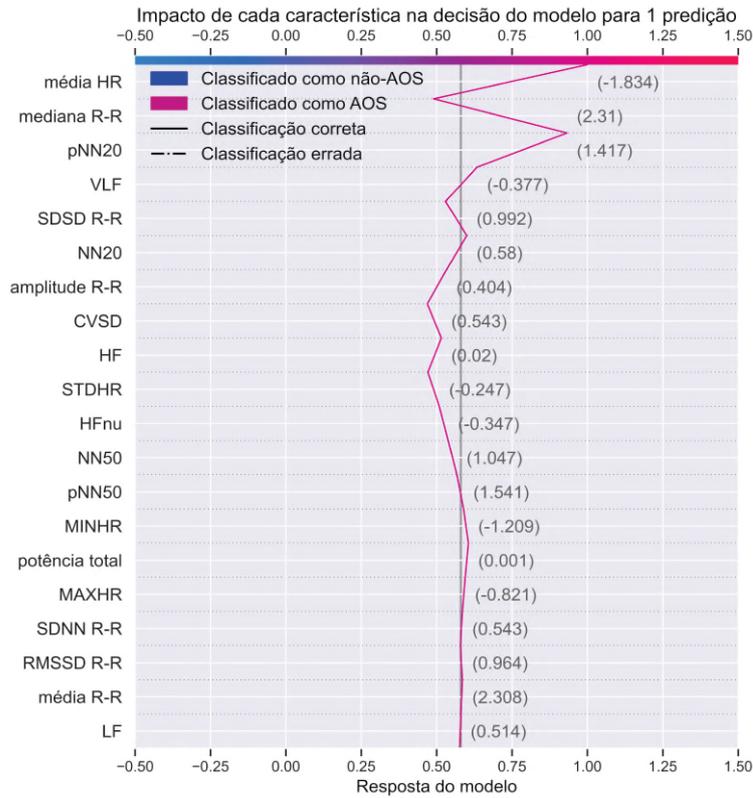
LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo AdaBoost.

Os valores SHAP permitem uma análise de como cada características influenciou a decisão do modelo. Esta análise é bastante útil, tanto para aumentar o conhecimento do problema, como para visualizar o peso de cada característica no acerto ou erro do modelo. Para facilitar a leitura deste tipo de análise, primeiro é apresentado um caso simples de uma única para predição na FIGURA 49. Na FIGURA, o eixo x representa a resposta do modelo, em que 0 significa um evento de não-AOS (indicado pela cor azul na FIGURA) e o valor igual a 1 como evento de AOS (indicados como vermelho na FIGURA). O eixo x da FIGURA foi estendido de

-0,50 até 1,50 para melhor visualizar o comportamento do gráfico. A linha contínua da FIGURA representa uma inferência realizada pelo modelo e pelo fato de ser contínua e não tracejada representa que o modelo acertou esta inferência, isto é, o modelo predisse um evento de AOS que de fato era um evento de AOS. Para entender como o modelo chegou a este resultado é preciso acompanhar a trajetória da linha que parte da parte inferior da FIGURA, em um valor próximo a 0,55 (indicado pela linha cinza tracejada), e segue até o topo da FIGURA, terminando no valor 1 (evento de AOS). Este é o valor do qual a parte a decisão do modelo e pode ser considerado uma espécie de viés do modelo. Se considerarmos que valores maiores 0,5 serão classificados como eventos de AOS e iguais ou abaixo de 0,5 classificados eventos de não-AOS, então pode-se dizer que o modelo é levemente inclinado a classificar uma entrada nula como evento de AOS. A medida que a linha caminha em direção ao topo da FIGURA, as características são avaliadas pelo modelo e é possível observar o efeito daquela variável naquela inferência do modelo, com os valores entre parênteses sendo o valor (normalizado) daquela variável. Seguindo a ordem de baixo para cima, as variáveis LF, média R-R e RMSSD R-R praticamente não alteraram o resultado do modelo, ou seja, o modelo continuou com o mesmo valor inicial. As características MAXHR, potência total e MINHR levemente adicionam ao resultado do modelo em direção ao valor 1, com o maior efeito sendo produzido pela característica potência total. Em seguida, a resposta do modelo é direcionada para o valor zero, com maior efeito causado pelas características STDHR e CVSD. Pode se dizer que para esta observação em específico, os valores destas características trouxeram a decisão do modelo para um evento de não-AOS, o que seria errado. As características NN20 e amplitude R-R aumentam o resultado em direção ao 1, com SDSD R-R diminuindo levemente. Então há um grande direcionamento com o valor de VLF e pNN20, sendo estas a quarta e terceira características, respectivamente, que mais contribuíram para o resultado do modelo desta observação específica, isso é observado pelo quanto a linha se inclina e se aproxima do 1,0. Finalmente, a característica mediana R-R diminui o resultado do modelo para então a característica média HR ser a que mais contribuiu para o resultado e direcionar o resultado para 1,0.

Em resumo, o gráfico da FIGURA 49 mostra no eixo y as características e no eixo x o resultado do modelo. A linha mostra a contribuição de cada características no resultado partindo de um valor inicial. Através da inclinação da linha no eixo x é possível analisar o quanto a característica contribuiu para a resposta do modelo. Além disso, as características estão organizadas das menores influências na parte inferior da FIGURA para as maiores na parte superior. O valor entre parênteses é o valor da característica, o que ajuda um especialista no domínio do problema a interpretar o efeito dos valores da característica, grandes ou pequenos, no processo decisório do modelo. Para o caso da FIGURA apresentada, os valores apresentados estão normalizados, representando

FIGURA 49 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DE UM EXEMPLO DO MODELO ADABOOST



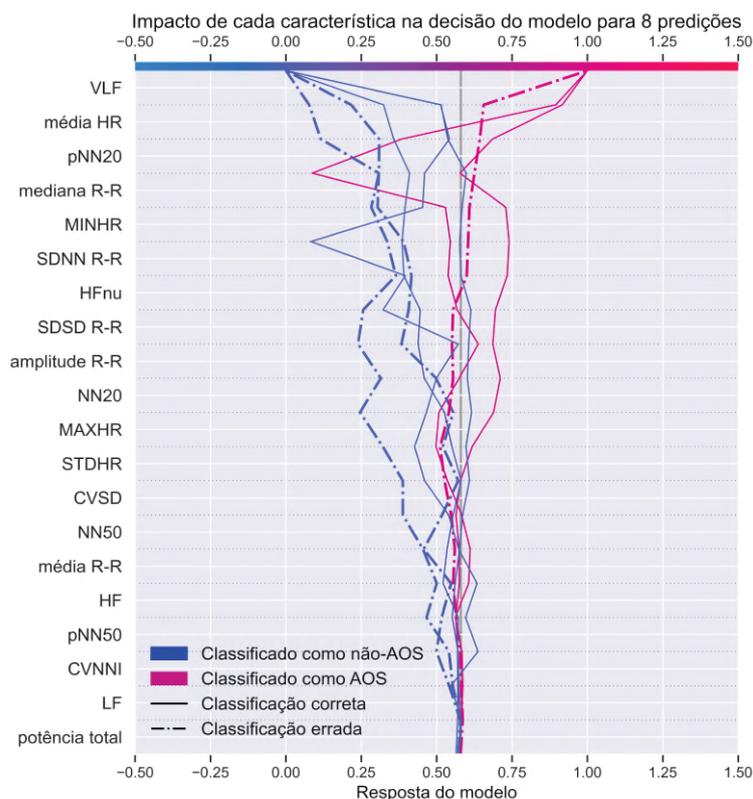
FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de um exemplo específico do modelo AdaBoost no experimento I.

mais o que foi de fato utilizado pelo modelo, mas que pode dificultar esta leitura de um especialista do domínio. Importante ressaltar que esta FIGURA trás apenas uma observação, e o *rank* das características é com base nesta observação e resultado em específico. Esta análise pode ser muito útil para entender com mais profundidade casos específicos. Mas limitadas em termos de comportamentos gerais do modelo. A FIGURA 50 ilustra uma análise com mais inferências, com um total de 8 exemplos (4 de cada classe) analisados.

O gráfico da FIGURA 50 tem a mesma estrutura do gráfico anterior (FIGURA 49), com cada linha azul ou rosa é uma inferência do modelo. As linhas azuis são as classificadas pelo modelo com eventos de não-AOS e as rosas como AOS. As linhas que são tracejadas indicam inferências erradas do modelo e as contínuas inferências corretas. Diferentemente da FIGURA anterior, como neste caso são apresentados múltiplas inferências, os valores de cada característica são omitidos. Mas a contribuição continua sendo possível de observar pela inclinação da linha no eixo x. A ordem das características está diferente, pois agora leva em conta média da importância de cada observação e não uma observação isolada. Em outras palavras, VLF, média

FIGURA 50 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DE OITO CASOS DO MODELO ADABOOST



FONTE: Próprio autor

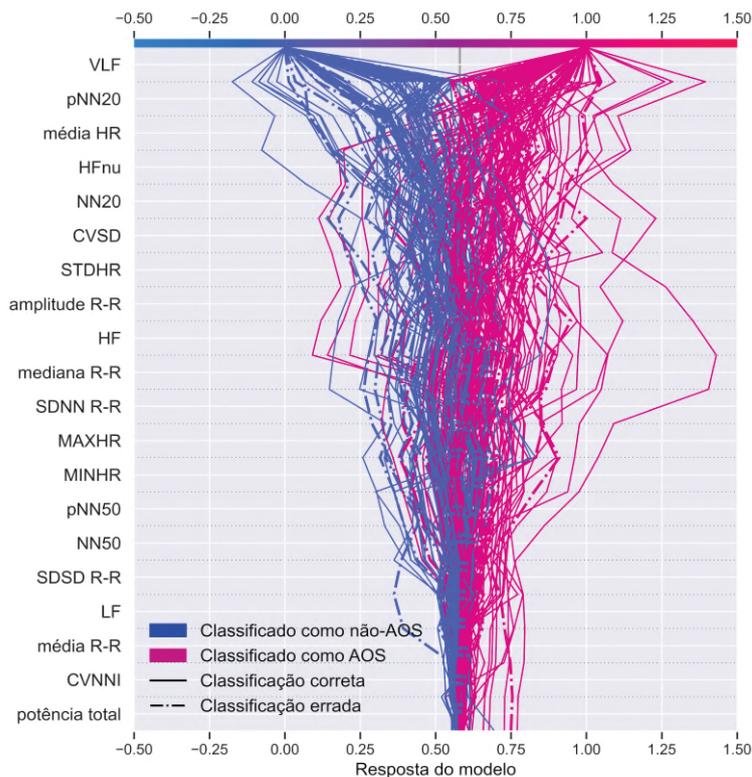
LEGENDA: Contribuição de cada característica no processo decisório da inferência de oito exemplos específico do modelo AdaBoost no experimento I.

HR, pNN20 e mediana R-R foram as características que, na média, mais explicaram o resultado destes oito exemplos. Para uma observação em específico, é possível analisar que STDHR e MAXHR tiveram impacto significativo, enquanto amplitude R-R, SDDSD R-R, HFnu e SDNN R-R tiveram impacto significativo em outra observação isolada. Potência total, LF e CVNNI foram as que menos tiveram impacto no resultado destas oito observações. Uma análise interessante na FIGURA 50 é o efeito das características nos acertos e erros do modelo. É possível afirmar que nestes oito exemplos as características MINHR e mediana R-R tiveram um efeito de direcionar o resultado para um evento de não-AOS, causando, inclusive, dois erros de falso negativo pelo modelo. A característica VLF teve um impacto significativo em um erro de falso positivo do modelo. Estas análises podem ajudar a entender padrões que o modelo capturou errado ou que não conseguiu capturar. Também servem de informações para o especialista do domínio como valores de características que causam uma interpretação errada do quadro ou que fogem do padrão.

Finalmente, a FIGURA 51 apresenta a mesma análise para todas as 200 amostras selecionadas. Da mesma forma da FIGURA 50, cada linha representa uma

inferência. Desta vez, é possível analisar de forma mais geral o comportamento do modelo. Assim como na FIGURA anterior, as características VLF e pNN20 foram as que mais contribuíram, na média, para os resultados do modelo. A análise mais geral do modelo permite observar alguns novos comportamentos, como alguns valores da característica pNN20, HF, SDNN R-R, amplitude R-R e mediana R-R tiveram impacto grande no resultado de algumas observações, praticamente tornando o resultado igual a 1 ou 0 (as vezes até mais que estes valores). Estes valores podem ser analisados em conjunto com um especialista do domínio e podem contribuir para melhor entender o problema. De forma similar, alguns valores das características que menos contribuíram para o resultado do modelo (potência total, CVNNI, média R-R e LF) resultaram um erro de falso positivo. Uma análise profunda destes valores pode ajudar no direcionamento de teste de novas hipóteses. Outra observação possível é que as características que ficaram no meio do eixo y (não sendo as que mais e nem as que menos contribuíram) tiveram impacto significativo no resultado de algumas observações. Isto pode conduzir estudos para melhor entender o papel destas características no processo decisório e no problema como um todo.

FIGURA 51 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO ADABOOST



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo Adaboost no experimento I.

Para os demais experimentos e demais modelos (com a exceção da SVM,

*bagging* e VM), foram gerados os gráficos de importância média de cada característica (FIGURA 48) e os gráficos do impacto na decisão do modelo para as 200 amostras (FIGURA 51) que se encontram no Anexo 1.

### 6.2.3 Resultados da Otimização dos hiperparâmetros e dos modelos

Esta seção apresenta os resultados da otimização, com os hiperparâmetros selecionados por cada modelo, bem como tempo da otimização e alguns outros resultados em subseções para cada um dos modelos. É dado um enfoque maior na discussão da SVM, modelo que obteve maior *F1-score* em todos os experimentos. Para os demais modelos, todos os resultados não apresentados Anexo 1.

#### 6.2.3.1 Máquinas de vetores de suporte (SVM)

A TABELA 39 apresenta os hiperparâmetros selecionados em cada um dos experimentos. Na TABELA, é possível observar que o *kernel* RBF apresentou melhor desempenho em todos os experimentos. O hiperparâmetro C tem como função controlar o balanceamento entre acuracidade da SVM e a simplicidade da forma da superfície de separação. Quanto menor o valor de C, mais simples é a superfície de separação entre as classes e quanto maior o valor, mais a SVM vai tentar classificar corretamente todos os exemplos. A medida que o número de dimensões foi diminuindo, o valor do hiperparâmetro C também diminuiu. Isto pode ter acontecido pelo fato de que, em uma menor dimensão, uma superfície de separação mais simples conseguia separar as classes. Com a exceção do experimento III, o tempo diminuiu à medida que o espaço de características diminuiu. No entanto, essa diminuição não foi significativa, sendo um pouco mais de 1 minuto na diminuição de 23 dimensões para 11 (experimento sem redução para experimento PCA) e de quase meio minuto na redução de 11 para 3 (experimento PCA para experimento UMAP). O valor do hiperparâmetro *gamma* não foi avaliado no estudo da otimização e foi definido como o inverso do número de características.

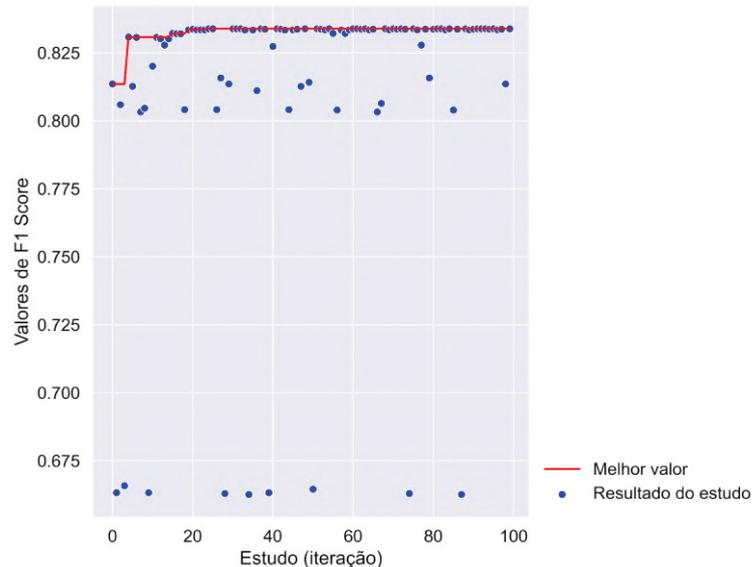
TABELA 39 – HIPERPARÂMETROS SELECIONADOS NO MODELO SVM E TEMPO DE OTIMIZAÇÃO

Experimento	Hiperparâmetro	Valor	Duração (mins)
1 - Sem redução	C	2,00	12,48
	Kernel	RBF	
2 - PCA	C	1,10	11,10
	Kernel	RBF	
3 - t-SNE	C	0,80	16,95
	Kernel	RBF	
4 - UMAP	C	0,50	10,78
	Kernel	RBF	

FONTE: Elaborada pelo autor

Na FIGURA 52 mostra o histórico de otimização da SVM com as bases de treinamento e validação. Na FIGURA, cada ponto representa um estudo (ou iteração) do processo de otimização, isto é, cada ponto é uma configuração de hiperparâmetros selecionados pelo Optuna para avaliar o desempenho, medido pelo *F1-score* no eixo y. A linha vermelha mostra o melhor valor obtido e como este valor evoluiu ao longo do processo. Na FIGURA é possível observar que, enquanto a maior parte das configurações obteve resultados entre 0,80 e 0,82, algumas configurações obtiveram valores bem inferiores, abaixo de 0,67. Isso mostra que alguns hiperparâmetros não possibilitaram a criação de superfícies que fossem capazes de separar as classes tão bem. Como a SVM é um modelo com poucos hiperparâmetros estudados (apenas dois), é possível realizar mais análises e entender melhor o comportamento deste modelo neste problema. Como um exemplo de aplicação e estudo, a TABELA 40 mostra a média dos resultados obtidos com os quatro *kernel* possíveis. Analisando a TABELA, é possível observar que o *kernel* do tipo sigmoide apresentou os piores resultados, o que mostra que a transformação dimensional utilizando como base a função sigmoide não conseguiu separar bem as classes, não captando algum tipo de padrão que os outros tipo de *kernel* conseguiram.

FIGURA 52 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO SVM NO EXPERIMENTO I



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de SVM no experimento I.

O histórico do processo de otimização é mostrado nas FIGURAS 53, 54 e 55 para os demais experimentos. No experimento com PCA (FIGURA 53), o comportamento foi similar ao do experimento sem redução, inclusive com resultados similares.

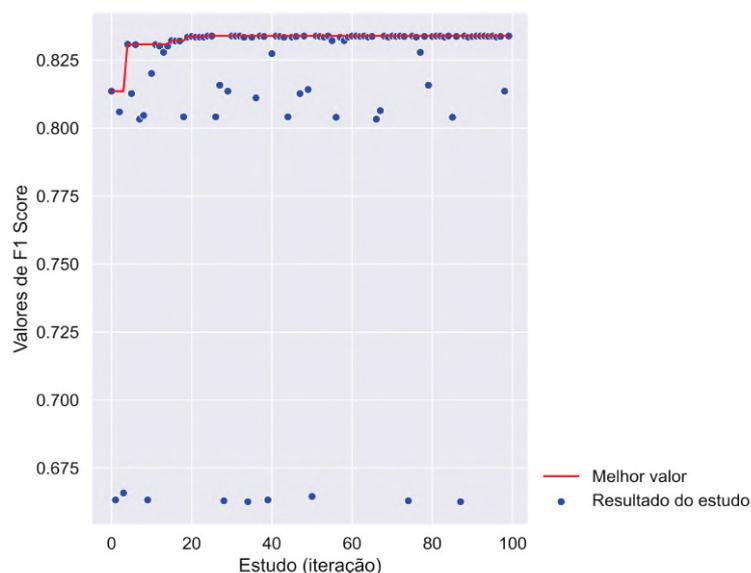
TABELA 40 – HIPERPARÂMETROS SELECIONADOS NO MODELO SVM E TEMPO DE OTIMIZAÇÃO

<i>Kernel</i>	<i>Média de F1-score</i>
RBF	0,83
Linear	0,80
Polinomial	0,81
Sigmoide	0,66

FONTE: Elaborada pelo autor

Na FIGURA 54 do experimento com t-SNE, é possível observar duas novas tendências aos gráficos dos experimentos anteriores, com algumas configurações chegando em resultados de aproximadamente 0,76 e outras em 0,74, com as melhores combinações estando próximas de 0,78 e as piores em cerca de 0,64. Finalmente, na FIGURA 55 do experimento com UMAP, três tendências são observadas, similar ao que acontece nos experimentos I e II, com a diferença de que as configurações de pior desempenho, tiveram um resultado inferior quando comparadas aos outros experimentos, com valores ao redor de 0,55.

FIGURA 53 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO SVM NO EXPERIMENTO II

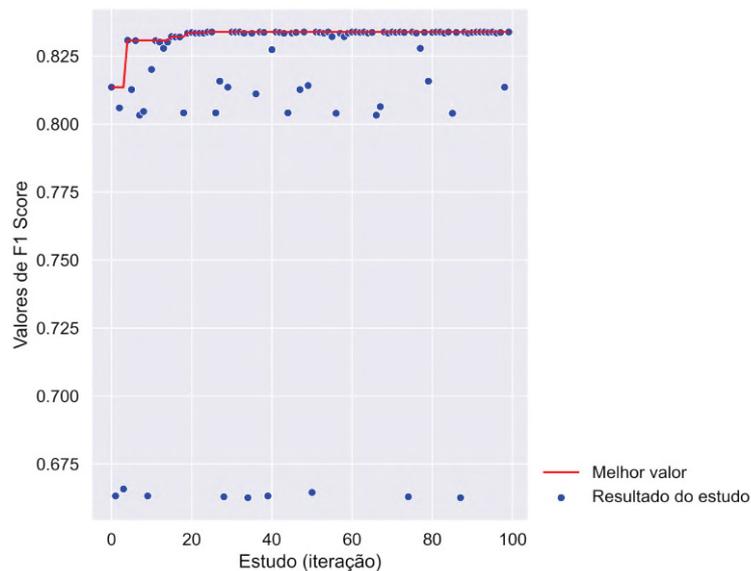


FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de SVM no experimento II.

Outra análise interessante é o impacto dos hiperparâmetros no desempenho e no tempo de otimização dos modelos durante a etapa de treinamento. A FIGURA 56 mostra estes dados para o modelo SVM. Na FIGURA 56, o gráfico da parte superior da FIGURA, indicado por "a", mostra o impacto relativo no resultado durante o treinamento, isto é, o quanto cada hiperparâmetro contribuiu para a classificação correta dos eventos de AOS e não-AOS. Como foram estudados apenas dois hiperparâmetros e conforme

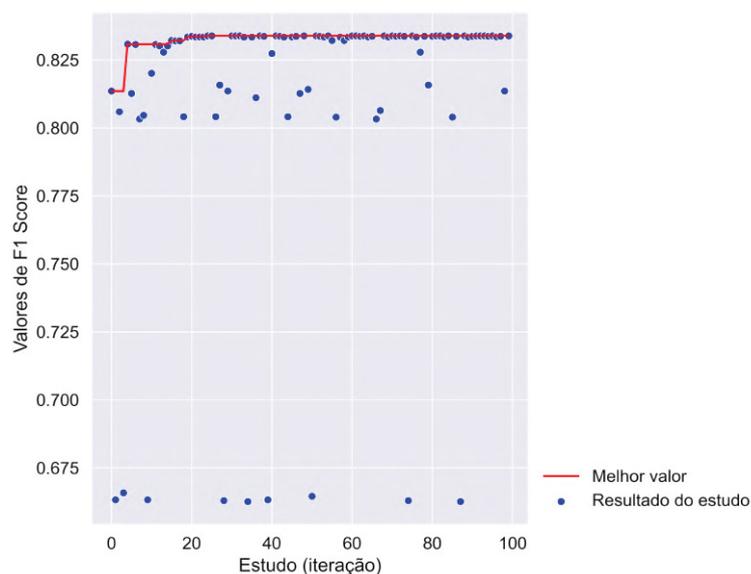
FIGURA 54 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO SVM NO EXPERIMENTO III



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de SVM no experimento III.

FIGURA 55 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO SVM NO EXPERIMENTO IV



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de SVM no experimento IV.

foi possível observar na análise da TABELA 40, o kernel foi praticamente responsável pelo desempenho do modelo em todos os experimentos.

O gráfico da parte inferior, "b", mostra o impacto relativo daquele hiperparâmetro no treinamento do modelo, isto é, o quanto que variando os valores daquele hiperparâmetro o tempo de treinamento aumentou ou diminuiu. Esta análise é útil, pois

em conjunto com o gráfico da direita, é possível observar quais hiperparâmetros vale a pena investigar num balanceamento entre resultado e tempo de treinamento. No caso da SVM, com a exceção do espaço transformado no experimento do UMAP, o *kernel* ainda foi o hiperparâmetro que mais impactou no tempo. No experimento do IV, o hiperparâmetro C foi responsável por mais impactar no treino, mas com pouca importância para desempenho do modelo. Portanto, o hiperparâmetro C poderia ser configurado com um valor padrão, enquanto se avalia quais *kernels* melhor ajudam a distinguir as duas classes.

A FIGURA 57 mostra a matriz de confusão para SVM em cada um dos experimentos. Na FIGURA, as diagonais de cada matriz correspondem aos acertos da SVM. É possível observar que, para SVM, um espaço de características maior possibilitou um maior acerto dos eventos de AOS, enquanto um espaço de característica menor possibilitou um maior acerto de eventos de não-AOS. Isso pode ajudar novas estratégias de modelagem, como utilização de *ensembles* com diferentes espaços de características.

Para finalizar os resultados da SVM, a TABELA 41 apresenta o tempo de predição e os resultados dos testes de hipóteses. O tempo de predição de toda a base de teste ficou em aproximadamente 14 segundos. Ambos os testes de hipóteses confirmaram que os resultados do modelo de SVM não foram provenientes do acaso, uma vez que ambos os p-valores ficaram abaixo do alfa definido (0,05).

TABELA 41 – TEMPOS DE TESTE E RESULTADO DOS TESTES DE HIPÓTESES DO MODELO SVM

Experimento	Tempo de predição (s)	p-Valor do teste binomial	p-Valor do teste McNema
I - Sem redução	14,53	6,37e-69	1,52e-23
II - PCA	12,95	1,90e-69	1,52e-23
III - t-SNE	13,24	3,77e-55	1,52e-23
IV - UMAP	14,35	6,74e-59	1,52e-23

FONTE: Elaborada pelo autor

### 6.2.3.2 *Adaptative Boosting*, AdaBoost

A TABELA 42 mostra os hiperparâmetros selecionados pelo Optuna para o modelo AdaBoost. No caso do AdaBoost, a diminuição da quantidade de dimensões impactou significativamente, reduzindo em cerca de 26 minutos de 23 para 3 dimensões. O hiperparâmetro SAMME.R foi selecionado três vezes. A taxa de aprendizado foi menor em espaços de maiores de dimensões e maior em espaços de menor dimensão, com o experimento III com o maior valor de taxa de aprendizado. Isso pode ter acontecido pelo fato de que, em espaços de dimensão elevadas, a complexidade dos padrões nos dados pode ser maior do que em espaços de menor dimensão, portanto a velocidade do aprendizado tende a ser menor para que o modelo possa capturar toda essa

complexidade a mais. Com a exceção do experimento do UMAP, que teve um número elevado de estimadores, o hiperparâmetro de total de estimadores diminuiu quanto menor foi o espaço de características e, portanto, menor a complexidade necessária para a superfície de separação pode ter sido menor.

TABELA 42 – HIPERPARÂMETROS SELECIONADOS NO MODELO SVM E TEMPO DE OTIMIZAÇÃO

Experimento	Hiperparâmetro	Valor	Duração (mins)
1 - Sem redução	Algoritmo	SAMME.R	31,10
	Taxa de aprendizado	0,20	
	N° de estimadores	693,00	
2 - PCA	Algoritmo	SAMME.R	17,48
	Taxa de aprendizado	0,70	
	N° de estimadores	320,00	
3 - t-SNE	Algoritmo	SAMME	3,79
	Taxa de aprendizado	1,60	
	N° de estimadores	140,00	
4 - UMAP	Algoritmo	SAMME.R	5,45
	Taxa de aprendizado	0,60	
	N° de estimadores	538,00	

FONTE: Elaborada pelo autor

### 6.2.3.3 Redes Neurais do tipo multicamadas de *perceptrons*, MLP

Os hiperparâmetros selecionados pelo modelo MLP variaram conforme o experimento, o que impactou também no tempo de otimização, conforme apresentado pela TABELA 43. Em questão de tempo de otimização total, dois fatores parecem ter contribuído, a estrutura da rede e o total de características. Experimentos II e III, que tem menores dimensões, tiveram maior tempo de duração devido a uma rede mais complexa, com mais camadas ou *perceptrons* (unidades) por camada, especialmente no caso do experimento III. Nos experimentos sem redução e UMAP, o tempo de processamento foi menor, no caso do sem redução com uma rede mais simples e no caso do UMAP, que possui uma rede mais complexa, mas um menor número de dimensões. Para 23 e 11 características, a função de ativação selecionada foi a logística, enquanto para três dimensões foi selecionado a função ReLU. Os experimentos com menor número de características, III e IV, maior foi o número de camadas selecionadas. Isso pode estar associado ao desenho de uma superfície de separação mais complexa por conta da redução de informação.

### 6.2.3.4 *Bagging*

Na TABELA 44 é mostrado os hiperparâmetros selecionados para o modelo *Bagging*. O processo de otimização demorou mais para o experimento II, mesmo com uma dimensão menor. A melhor configuração encontrada pelo Optuna no experimento

TABELA 43 – HIPERPARÂMETROS SELECIONADOS NO MODELO MLP E TEMPO DE OTIMIZAÇÃO

Experimento	Hiperparâmetro	Valor	Duração (mins)
1 - Sem redução	Número de camadas	1,00	8,28
	Unidades na camada 1	42,00	
	Função ativação	Logística	
	Alfa	$1,00 \times 10^4$	
	Taxa de aprendizado	0,01	
2 - PCA	Número de camadas	1,00	9,51
	Unidades na camada 1	52,00	
	Função ativação	Logística	
	Alfa	$6,00 \times 10^4$	
	Taxa de aprendizado	0,01	
3 - t-SNE	Número de camadas	4,00	17,59
	Unidades na camada 1	92,00	
	Unidades na camada 2	96,00	
	Unidades na camada 3	100,00	
	Unidades na camada 4	21,00	
	Função ativação	ReLU	
	Alfa	$1,00 \times 10^4$	
	Taxa de aprendizado	0,01	
	Número de camadas	3,00	
	Unidades na camada 1	65,00	
4 - UMAP	Unidades na camada 2	41,00	5,76
	Unidades na camada 3	45,00	
	Função ativação	ReLU	
	Alfa	$1,00 \times 10^4$	
	Taxa de aprendizado	0,01	

FONTE: Elaborada pelo autor

sem redução de dimensão foi com um classificador do tipo DT e uma quantidade elevada de classificadores. Esta configuração foi similar ao experimento do PCA, em que o classificador escolhido também foi uma DT, mas o número de classificadores caiu em mais de 2,5 vezes, isso pode ter ocorrido pela diminuição de dimensões no espaço de características. Em ambos os experimentos do t-SNE e UMAP, o classificador escolhido foi o KNN, com diferença para o número de classificadores, em que no experimento UMAP foi mais que o dobro do que no experimento t-SNE.

#### 6.2.3.5 Árvores de decisão, DT

Novamente, o efeito da diminuição de características ajudou em um menor tempo de otimização, como é possível observar na TABELA 45. A diminuição foi de cerca de 6 minutos no experimento I com 23 características para 11 no experimento. E de cerca de 11 minutos do experimento II para os experimentos III e IV, ambos com três características. As melhores configurações para experimento sem redução de dimensão e com PCA foram com a entropia como critério de divisão, enquanto para os experimentos t-SNE e UMAP o critério selecionado foi o índice de Gini. Todos os

TABELA 44 – HIPERPARÂMETROS SELECIONADOS NO MODELO *BAGGING* E TEMPO DE OTIMIZAÇÃO

Experimento	Hiperparâmetro	Valor	Duração (mins)
1 - Sem redução	Classificador	DT	182,06
	N° de classificadores	648,00	
	Razão de amostras	0,34	
	Reposição	Falso	
2 - PCA	Classificador	DT	195,80
	N° de classificadores	241,00	
	Razão de amostras	0,61	
	Reposição	Falso	
3 - t-SNE	Classificador	KNN	42,19
	N° de classificadores	203,00	
	Razão de amostras	0,40	
	Reposição	Falso	
4 - UMAP	Classificador	KNN	39,34
	N° de classificadores	421,00	
	Razão de amostras	0,24	
	Reposição	Verdadeiro	

FONTE: Elaborada pelo autor

experimentos utilizaram da estratégia de divisão "melhor", enquanto a profundidade máxima variou, sendo o experimento I com maior valor.

TABELA 45 – HIPERPARÂMETROS SELECIONADOS NO MODELO DT E TEMPO DE OTIMIZAÇÃO

Experimento	Hiperparâmetro	Valor	Duração (mins)
1 - Sem redução	Critério	Entropia	21,01
	Estratégia de divisão	Melhor	
	Profundidade máxima	10,00	
	Mínimo de amostras	3,00	
2 - PCA	Critério	Entropia	15,44
	Estratégia de divisão	Melhor	
	Profundidade máxima	8,00	
	Mínimo de amostras	4,00	
3 - t-SNE	Critério	Gini	4,66
	Estratégia de divisão	Melhor	
	Profundidade máxima	9,00	
	Mínimo de amostras	4,00	
4 - UMAP	Critério	Gini	3,50
	Estratégia de divisão	Melhor	
	Profundidade máxima	4,00	
	Mínimo de amostras	4,00	

FONTE: Elaborada pelo autor

#### 6.2.3.6 *Ensemble* com voto majoritário, VM

Para melhor leitura dos hiperparâmetros selecionados pelo modelo VM, os resultados foram divididos em quatro diferentes, uma para cada experimento. A TABELA 46 mostra os hiperparâmetros selecionados para o experimento I, que durou

379,48 minutos. Ao todo foram selecionados 6 estimadores, sendo 3 DTs e 3 SVMs. As DTs tiveram uma profundidade média de aproximadamente 7, enquanto os valores dos hiperparâmetros da SVM foram bastante elevados.

TABELA 46 – HIPERPARÂMETROS SELECIONADOS NO MODELO VM E TEMPO DE OTIMIZAÇÃO NO EXPERIMENTO I

Hiperparâmetro	Valor
N° de estimadores	3,00
Profundidade da árvore 1	9,00
C da SVM 2	93756,31
Gama da SVM 1	48879,61
Profundidade da árvore 2	7,00
C da SVM 2	7896,60
Gama da SVM 2	7873,04
Profundidade da árvore 3	6,00
C da SVM 3	51232,21
Gama da SVM 3	44921,14
Peso árvore 1	0,38
Peso SVM 1	0,85
Peso árvore 2	0,91
Peso SVM 2	0,77
Peso árvore 3	0,68
Peso SVM 3	0,14

FONTE: Elaborada pelo autor

A TABELA 47 mostra os hiperparâmetros selecionados no experimento com PCA. Foram selecionados 2 estimadores que resultam em um total de 4 classificadores, duas SVM e duas DT. Para a DT a profundidade máxima média ficou em aproximadamente 8, enquanto os hiperparâmetros da SVM continuaram elevados. O tempo total de execução da otimização deste modelo foi de 257,13 minutos.

TABELA 47 – HIPERPARÂMETROS SELECIONADOS NO MODELO VM E TEMPO DE OTIMIZAÇÃO NO EXPERIMENTO II

Hiperparâmetro	Valor
N° de estimadores	2,00
Profundidade da árvore 1	6,00
C da SVM 2	32372,44
Gama da SVM 1	6643,31
Profundidade da árvore 2	9,00
C da SVM 2	17198,98
Gama da SVM 2	73819,20
Peso árvore 1	0,33
Peso SVM 1	0,51
Peso árvore 2	0,81
Peso SVM 2	0,43

FONTE: Elaborada pelo autor

Para o experimento III, foram selecionados quatro estimadores. Os resultados são mostrados na TABELA 48. A estrutura das DTs variou, com as duas primeiras possuindo maior profundidade do que as duas últimas, na média a profundidade

máxima foi de 7. O comportamento da SVM segue similar aos experimentos anteriores. O tempo de otimização total foi de 378,90 minutos, o mais elevado dos experimentos. Isso pode ter acontecido pelas configurações com um maior número de estimadores utilizados.

TABELA 48 – HIPERPARÂMETROS SELECIONADOS NO MODELO VM E TEMPO DE OTIMIZAÇÃO NO EXPERIMENTO III

Hiperparâmetro	Valor
Nº de estimadores	4,00
Profundidade da árvore 1	9,00
C da SVM 2	5431,55
Gama da SVM 1	6310,77
Profundidade da árvore 2	8,00
C da SVM 2	53348,39
Gama da SVM 2	82617,07
Profundidade da árvore 3	3,00
C da SVM 3	20019,56
Gama da SVM 3	15519,74
Profundidade da árvore 4	6,00
C da SVM 4	10967,17
Gama da SVM 4	45849,56
Peso árvore 1	0,56
Peso SVM 1	0,50
Peso árvore 2	0,82
Peso SVM 2	0,05
Peso árvore 3	0,36
Peso SVM 3	0,14
Peso árvore 4	0,65
Peso SVM 4	0,02

FONTE: Elaborada pelo autor

A TABELA 49 apresenta os resultados para resultado do experimento com UMAP, que levou 370,29 minutos para otimização total. Quatro estimadores foram selecionados, assim como no experimento do t-SNE. O comportamento da estrutura das DTs ficou mais variado do que no experimento com t-SNE. Os pesos dos modelos ficaram equilibrados, em comparação com os demais modelos.

#### 6.2.3.7 *Light gradient boosting machine*, LightGBM

Para o modelo LightGBM, os resultados são apresentados em duas TABELAS, a TABELA 50 e a TABELA 51. Na TABELA 50, é possível observar que no experimento sem redução de dimensão, a melhor configuração encontrada contém menos classificadores, mas com mais cada classificador com mais folhas, do que ocorre no experimento com PCA. Nos experimentos t-SNE e UMAP (TABELA 51), tanto o número de estimadores, quanto número de folhas diminuiu. A otimização do LightGBM foi extremamente rápida para os experimentos com menos dimensões.

TABELA 49 – HIPERPARÂMETROS SELECIONADOS NO MODELO VM E TEMPO DE OTIMIZAÇÃO NO EXPERIMENTO IV

Hiperparâmetro	Valor
N° de estimadores	4,00
Profundidade da árvore 1	4,00
C da SVM 2	85624,13
Gama da SVM 1	54775,21
Profundidade da árvore 2	7,00
C da SVM 2	58861,08
Gama da SVM 2	30564,88
Profundidade da árvore 3	3,00
C da SVM 3	37967,86
Gama da SVM 3	28436,17
Profundidade da árvore 4	6,00
C da SVM 4	24696,43
Gama da SVM 4	19133,43
Peso árvore 1	0,77
Peso SVM 1	0,58
Peso árvore 2	0,83
Peso SVM 2	0,32
Peso árvore 3	0,94
Peso SVM 3	0,46
Peso árvore 4	0,45
Peso SVM 4	0,55

FONTE: Elaborada pelo autor

TABELA 50 – HIPERPARÂMETROS SELECIONADOS NO MODELO LIGHTGBM E TEMPO DE OTIMIZAÇÃO NO EXPERIMENTO I E II

Experimento	Hiperparâmetro	Valor	Duração (mins)
1 - Sem redução	N° de folhas	236,00	1,22
	Taxa de aprendizado	0,07	
	Profundidade máxima	13,00	
	Soma mínima de pesos	14,00	
	Razão de amostragem para instância de treinamento	0,18	
	Razão amostras de coluna	0,81	
	Regulador alfa	3,26	
	Regulador lambda	5,41	
	N° de classificadores	158,00	
	N° de folhas	115,00	
2 - PCA	Taxa de aprendizado	0,06	1,12
	Profundidade máxima	13,00	
	Soma mínima de pesos	23,00	
	Razão de amostragem para instância de treinamento	0,55	
	Razão amostras de coluna	1,00	
	Regulador alfa	7,07	
	Regulador lambda	9,08	
N° de classificadores	478,00		

FONTE: Elaborada pelo autor

### 6.2.3.8 Regressão logística, LR

A TABELA 52 mostra as melhores configurações para modelo LR. Na TABELA, é possível observar que, apesar do tempo pequeno de otimização para as 23 características do experimento I, este tempo é ainda menor à medida que as dimensões

TABELA 51 – HIPERPARÂMETROS SELECIONADOS NO MODELO LIGHTGBM E TEMPO DE OTIMIZAÇÃO NO EXPERIMENTO III E IV

Experimento	Hiperparâmetro	Valor	Duração (mins)
3 - t-SNE	N° de folhas	141,00	0,45
	Taxa de aprendizado	0,10	
	Profundidade máxima	15,00	
	Soma mínima de pesos	41,00	
	Razão de amostragem para instância de treinamento	0,47	
	Razão amostras de coluna	0,84	
	Regulador alfa	4,44	
	Regulador lambda	2,70	
	N° de classificadores	107,00	
	N° de folhas	123,00	
4 - UMAP	Taxa de aprendizado	0,03	0,50
	Profundidade máxima	9,00	
	Soma mínima de pesos	6,00	
	Razão de amostragem para instância de treinamento	0,45	
	Razão amostras de coluna	0,88	
	Regulador alfa	1,17	
	Regulador lambda	4,21	
	N° de classificadores	90,00	

FONTE: Elaborada pelo autor

diminuem, mas com um limiar em cerca de 0,15 minutos. Em todos os experimentos, a penalidade do tipo L1 foi selecionada, com valor do peso da penalidade e do hiperparâmetro C variando em todos os experimentos.

TABELA 52 – HIPERPARÂMETROS SELECIONADOS NO MODELO LR E TEMPO DE OTIMIZAÇÃO

Experimento	Hiperparâmetro	Valor	Duração (mins)
1 - Sem redução	Penalidade	L1	1,64
	C	8,25	
	Peso da penalidade "L1"	0,28	
2 - PCA	Penalidade	L1	0,15
	C	0,47	
	Peso da penalidade "L1"	0,25	
3 - t-SNE	Penalidade	L1	0,14
	C	1,48	
	Peso da penalidade "L1"	0,10	
4 - UMAP	Penalidade	L1	0,16
	C	2,48	
	Peso da penalidade "L1"	0,54	

FONTE: Elaborada pelo autor

### 6.2.3.9 Floresta aleatória, RF

Para o modelo RF, o experimento utilizando PCA foi que mais levou tempo de otimização, conforme é possível analisar na TABELA 53. Depois do experimento com PCA, é possível que a redução das dimensões diminuiu o tempo, se comparamos os experimentos sem redução de dimensão com os experimentos t-SNE e UMAP. A quantidade de estimadores foi elevada para os experimentos I, II e III, com uma diminuição

apenas no experimento IV. Apenas o experimento I teve como critério de separação a entropia, os demais experimentos a melhor configuração deste hiperparâmetro foi o índice de Gini.

TABELA 53 – HIPERPARÂMETROS SELECIONADOS NO MODELO RF E TEMPO DE OTIMIZAÇÃO

Experimento	Hiperparâmetro	Valor	Duração (mins)
1 - Sem redução	N° de estimadores	716.00	32.51
	Critério	Entropia	
	Profundidade máxima	28.00	
	Mínimo de amostras	4.00	
2 - PCA	N° de estimadores	854.00	36.69
	Critério	Gini	
	Profundidade máxima	32.00	
	Mínimo de amostras	3.00	
3 - t-SNE	N° de estimadores	691.00	16.68
	Critério	Gini	
	Profundidade máxima	19.00	
	Mínimo de amostras	8.00	
4 - UMAP	N° de estimadores	342.00	10.83
	Critério	Gini	
	Profundidade máxima	22.00	
	Mínimo de amostras	6.00	

FONTE: Elaborada pelo autor

#### 6.2.3.10 *Extreme gradient boosting*, xGBoost

Na TABELA 54 é apresentado os resultados para o modelo xGBoost. A melhor configuração do experimento com t-SNE utilizou a maior quantidade de estimadores, enquanto o experimento aplicando UMAP utilizou a menor quantidade. O tempo de duração da otimização também diminuiu à medida que o número de dimensões diminuiu. De quase 130 minutos com todas as características para cerca de 15 minutos no experimento com t-SNE ou cerca de 8 minutos no experimento com UMAP.

### 6.3 RESULTADOS DOS MODELOS DE APRENDIZADO PROFUNDO

De forma semelhante ao apresentado na seção de resultados dos modelos de ML (6.2), os resultados gerais são apresentados e depois os resultados da CNN (que obteve maior F1-score) são detalhados nas seções seguintes. Os demais resultados podem ser encontrado no apêndice 1.

#### 6.3.1 RESULTADOS GERAIS

A TABELA 55 apresenta os resultados de desempenho dos modelos de DL. Na média, os modelos de DL apresentaram um F1-score de 0,778, com o maior valor obtido pelo modelo do tipo CNN, que também obteve o maior acuracidade e precisão.

TABELA 54 – HIPERPARÂMETROS SELECIONADOS NO MODELO XGBOOST E TEMPO DE OTIMIZAÇÃO

Experimento	Hiperparâmetro	Valor	Duração (mins)
1 - Sem redução	Taxa de aprendizado	0,10	129,38
	N° de estimadores	697,00	
	Profundidade máxima	0,88	
	Razão de amostragem para instância de treinamento	9,00	
2 - PCA	Taxa de aprendizado	0,10	35,85
	N° de estimadores	474,00	
	Profundidade máxima	0,29	
	Razão de amostragem para instância de treinamento	10,00	
3 - t-SNE	Taxa de aprendizado	0,10	15,19
	N° de estimadores	825,00	
	Profundidade máxima	0,57	
	Razão de amostragem para instância de treinamento	3,00	
4 - UMAP	Taxa de aprendizado	0,10	8,43
	N° de estimadores	31,00	
	Profundidade máxima	0,87	
	Razão de amostragem para instância de treinamento	6,00	

FONTE: Elaborada pelo autor

De modo geral, os resultados ficaram próximos, com modelo híbrido alcançando um resultado muito próximo ao do modelo do tipo CNN. Isso pode ter acontecido pois modelos convolucionais extraem as características das própria entradas, ajudando na distinção das classes.

TABELA 55 – RESULTADOS DE DESEMPENHO DOS MODELOS DE DL

Modelo	Acuracidade	Recall	Precisão	F1-score
CNN	<b>0,752</b>	0,746	<b>0,844</b>	<b>0,792</b>
LSTM	0,705	<b>0,785</b>	0,722	0,752
Híbrido	0,728	0,758	0,824	0,790

FONTE: Elaborada pelo autor

Na TABELA 56 são mostrados os tempos de otimização e de predição e os resultados nos testes de hipóteses, executados nas máquinas descritas no capítulo 5. O modelo híbrido, que combina ambos os aspectos dos demais modelos de DL estudado, foi que teve a otimização mais demorada. Como neste modelo, foi fixado três camadas convolucionais, isto pode ter ajudado a aumentar o tempo, além das outras camadas de LSTM para serem sintonizadas. O modelo CNN e LSTM tiveram tempos de otimização semelhantes. O tempo de predição também foi maior para a rede híbrida maior e mais complexa levando cerca de 2 segundos, enquanto as demais ficaram com tempo abaixo de dois segundos, mas acima de 1 segundo. Todos os p-valores ficaram abaixo do limite testado ( $\alpha < 0,005$ ), portando a hipótese dos resultados obtidos serem frutos do mero acaso é descartada.

TABELA 56 – OUTROS RESULTADOS DOS MODELOS DE DL

Modelo	Tempo de otimização (min)	Tempo de predição (s)	p-valor do teste binominal	p-valor do teste McNemna
CNN	89,02	1,40	$1,16^{-47}$	$1,52^{-23}$
LSTM	90,62	1,68	$2,84 \times 10^{-39}$	$1,52 \times 10^{-23}$
Híbrido	141,38	2,02	$6,16 \times 10^{-49}$	$1,53 \times 10^{-23}$

FONTE: Elaborada pelo autor

### 6.3.2 Resultados da otimização

Esta seção apresenta os melhores hiperparâmetros encontrados para os modelos de DL e o impacto de cada um no resultado e nos tempos de otimização. Esta subseção é dividida para cada um dos modelos, de modo similar como foi apresentado para os modelos de ML na seção 6.2.3. Do mesmo modo, todos os resultados do melhor modelo obtido, CNN, serão apresentados nesta seção, enquanto os demais modelos o impacto de cada hiperparâmetro é discutido no 1.

#### 6.3.2.1 Redes neurais convolucionais, CNN

Os hiperparâmetros selecionados para o modelo CNN são apresentados na TABELA 57. A melhor configuração encontrada foi com 3 camadas no total, com a primeira e a segunda camada com uma função de convolução cada e a terceira com duas funções convolucionais. O tamanho do filtro foi o mesmo para todas as camadas, com 64, com o tamanho do *kernel* sendo maior na primeira camada, com valor de 7 e menor na segunda e na terceira camada, com 5. A função de ativação ReLU foi utilizada pelas camadas 2, 3 e densa, enquanto a função de ativação linear foi utilizada na camada 1.

A FIGURA 58 apresenta o histórico de otimização do modelo de CNN na base de validação. Nesta FIGURA, cada ponto é uma configuração testada com valor de F1-score obtido. A linha vermelha indica como o resultado da melhor configuração evoluiu ao longo da otimização. A maior parte das configurações obteve valores entre 0,75 e 0,80 de F1-score, com algumas configurações com resultado abaixo de 0,75. A pior configuração testada obteve cerca de 0,60 de F1-score.

A FIGURA 59 mostra o impacto relativo de cada hiperparâmetro no tempo e no resultado da otimização do modelo CNN. NO gráfico da parte superior da FIGURA 59, indicado por "a" é mostrado o impacto relativo ao desempenho. Os dois hiperparâmetros mais importante no resultado foram os que configuravam a camada densa da rede, o tamanho da camada densa como que mais impactou e o tipo de função de ativação da camada densa em segundo. O peso da arquitetura, controlado pelo hiperparâmetro número de camadas convolucionais, foi apenas o quarto hiperparâmetro mais importante no resultado. A primeira camada convolucional teve mais impacto que

TABELA 57 – HIPERPARÂMETROS SELECIONADOS PARA O MODELO CNN

Hiperparâmetro	Valor
N° de camadas convolucionais	3
N° de funções de convolução na camada 1	1
Tamanho do filtro na camada 1	64
Tamanho do <i>kernel</i> na camada 1	7
Função ativação na camada 1	Linear
Tamanho do passo na camada 1	2
N° de funções de convolução na camada 2	1
Tamanho do filtro na camada 2	64
Tamanho do <i>kernel</i> na camada 2	5
Função ativação na camada 2	ReLU
Tamanho do passo na camada 2	1
N° de funções de convolução na camada 3	2
Tamanho do filtro na camada 3	64
Tamanho do <i>kernel</i> na camada 3	5
Função ativação na camada 3	ReLU
Tamanho do passo na camada 3	1
N° de unidades na camada densa	128
Função ativação na camada densa	ReLU

FONTE: Elaborada pelo autor

as demais no resultado do modelo. No gráfico da parte inferior da FIGURA 59, indicado por "b" mostra o impacto no tempo. Os dois hiperparâmetros que mais impactaram no tempo foram o número de camadas convolucionais, que configura o tamanho da rede, portanto podendo construir redes que demandam mais tempo para treinamento e o hiperparâmetro de total de funções de convolução na camada 1, que aumenta ou diminui o tamanho total da camada 1, com a operação de convolução.

### 6.3.2.2 Redes neurais recorrentes, LSTM

A TABELA 58 apresenta os hiperparâmetros selecionados para o modelo de LSTM. A melhor configuração encontrada é de uma rede de complexidade menor, com apenas uma camada de LSTM, com 37 unidades e taxa de *dropout* de 0,2. A camada densa desta configuração também apresentou uma complexidade menor, com 16 unidades e função de ativação ReLU.

Os demais resultados são apresentados no 1

TABELA 58 – HIPERPARÂMETROS SELECIONADOS PARA O MODELO LSTM

Hiperparâmetro	Valor
N° de camadas de LSTM	1
Taxa de <i>dropout</i> na camada 1	0.2
N° de unidades na camada 1	37
N° de unidades na camada densa	16
Função ativação na camada densa	ReLU

FONTE: Elaborada pelo autor

### 6.3.2.3 Redes neurais híbridas de CNN e LSTM

A melhor configuração do modelo híbrido de CNN e LSTM também selecionou apenas 1 camada de LSTM, conforme mostrado na TABELA 59. Os filtros das três camadas de CNN apresentaram tamanhos elevados, de 512 na primeira e segunda camada e 128 na terceira camada. A normalização foi aplicada nas camadas de CNN. Todas as funções de ativação foram ReLU, com a exceção da camada densa em que a função de ativação linear foi escolhida. Por fim, o tamanho da camada de densa foi de 32.

TABELA 59 – HIPERPARÂMETROS SELECIONADOS PARA O MODELO HÍBRIDO DE CNN E LSTM

Hiperparâmetro	Valor
N° de camadas de LSTM	1
Aplicar normalização	Sim
Tamanho do filtro da camada de CNN 1	512
Função de ativação da camada de CNN 1	ReLU
Tamanho do filtro da camada de CNN 2	512
Função de ativação da camada de CNN 2	ReLU
Tamanho do filtro da camada de CNN 3	128
Função de ativação da camada de CNN 3	ReLU
Taxa de <i>dropout</i> da camada 1 de LSTM	0,30
N° de unidades da camada 1 de LSTM*	32
N° de unidades da camada densa	32
Função de ativação da camada densa	Linear

FONTE: Elaborada pelo autor

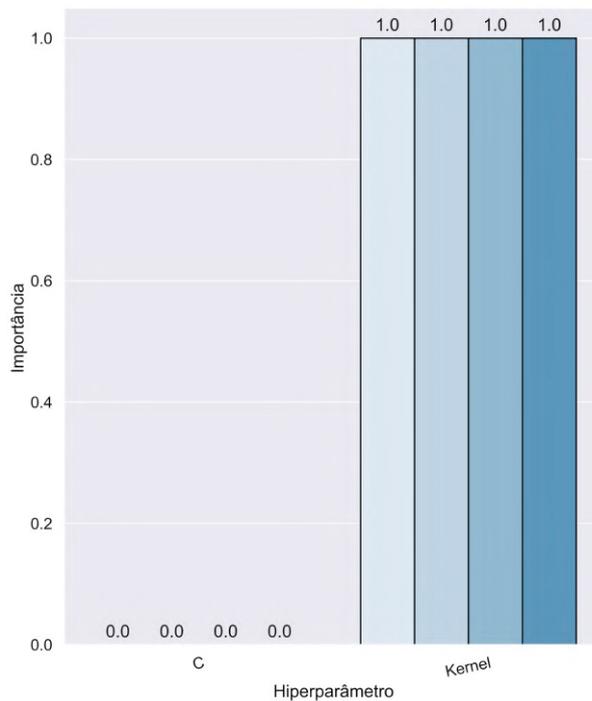
NOTA: \*Hiperparâmetro fixo baseado no valor do tamanho do filtro da camada 3 de CNN

De forma semelhante, os demais resultados da otimização são apresentados na capítulo 1

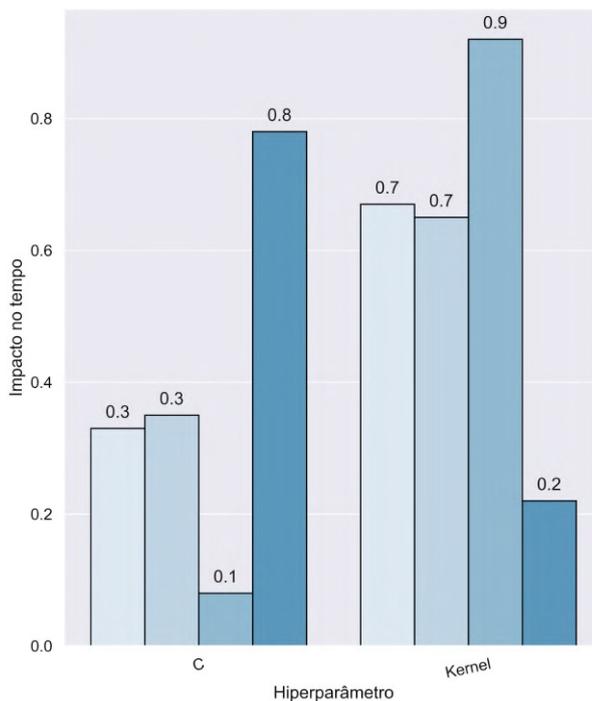
## 6.4 RESUMO

Este capítulo apresentou os resultados dos modelos de ML e DL. Os modelos de ML obtiveram melhores resultados, com destaque para SVM, melhor resultado obtido neste trabalho. Dentro os modelos de DL, CNN apresentou o melhor resultado. Foram apresentados os resultados da otimização de cada modelo, com destaque maior para SVM e CNN. Os resultados de explicabilidade foram apresentados com AdaBoost, uma vez que não foram realizados estudos de explicabilidade com a SVM. O próximo capítulo aprofunda as discussões dos resultados obtidos e compara com outros trabalhos.

FIGURA 56 – IMPACTOS DOS HIPERPARÂMETROS NA OTIMIZAÇÃO DO MODELO SVM



(a) importância relativa no desempenho

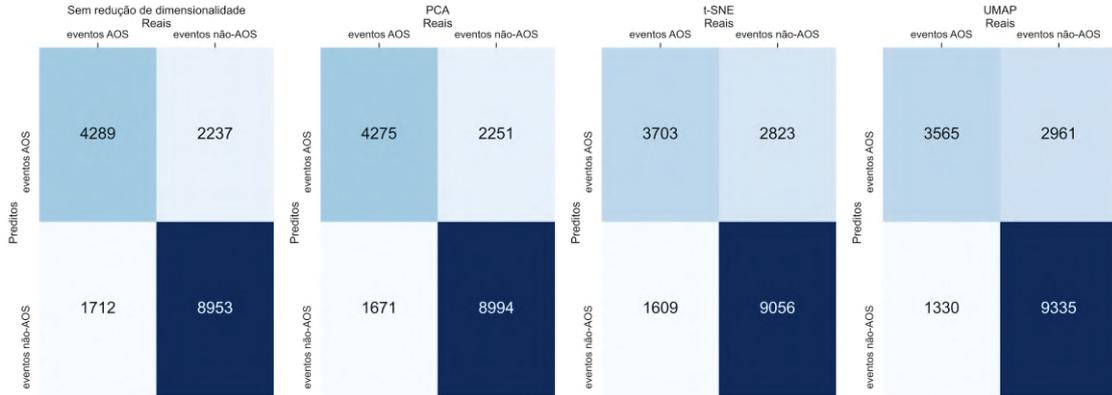


(b) impacto relativo no tempo de otimização

- Experimentos
- sem redução de dimensionalidade
  - PCA
  - t-SNE
  - UMAP

FONTE: Próprio autor  
 LEGENDA: Impactos dos hiperparâmetros no desempenho e no tempo de otimização para modelo SVM.

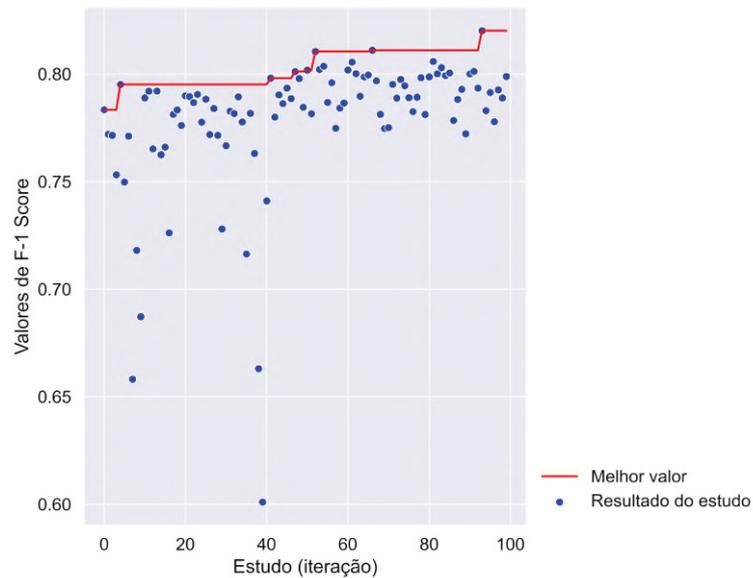
FIGURA 57 – MATRIZ DE CONFUSÃO DO MODELO SVM



FONTE: Próprio autor

LEGENDA: Matriz de confusão para o modelo SVM em cada um dos experimentos.

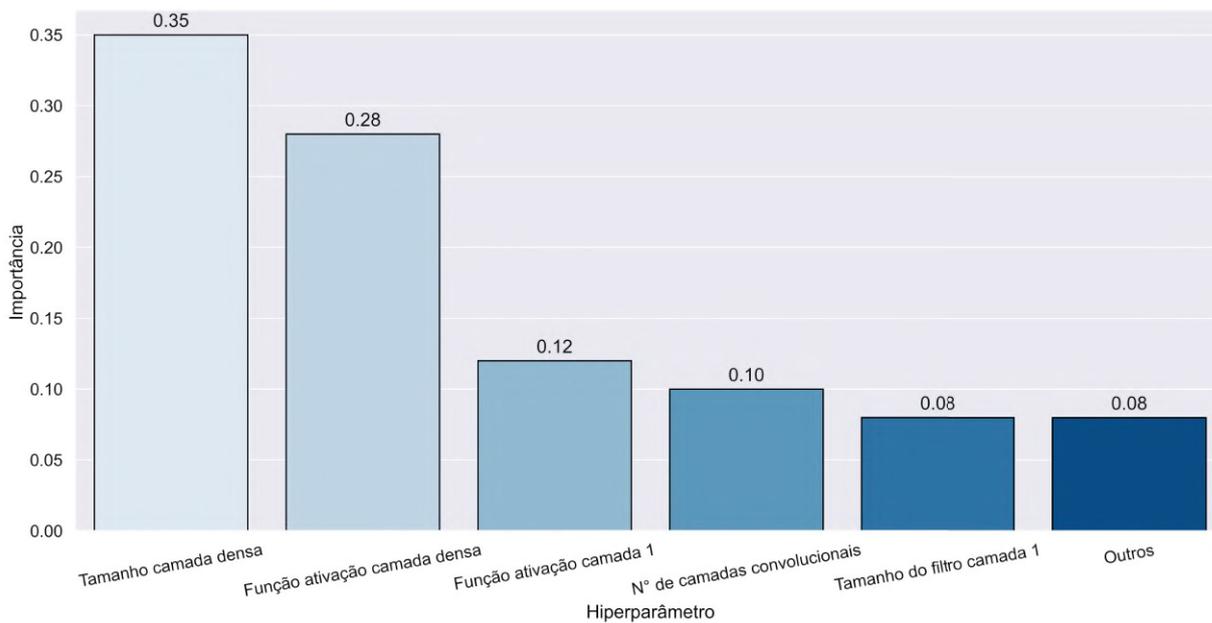
FIGURA 58 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO CNN



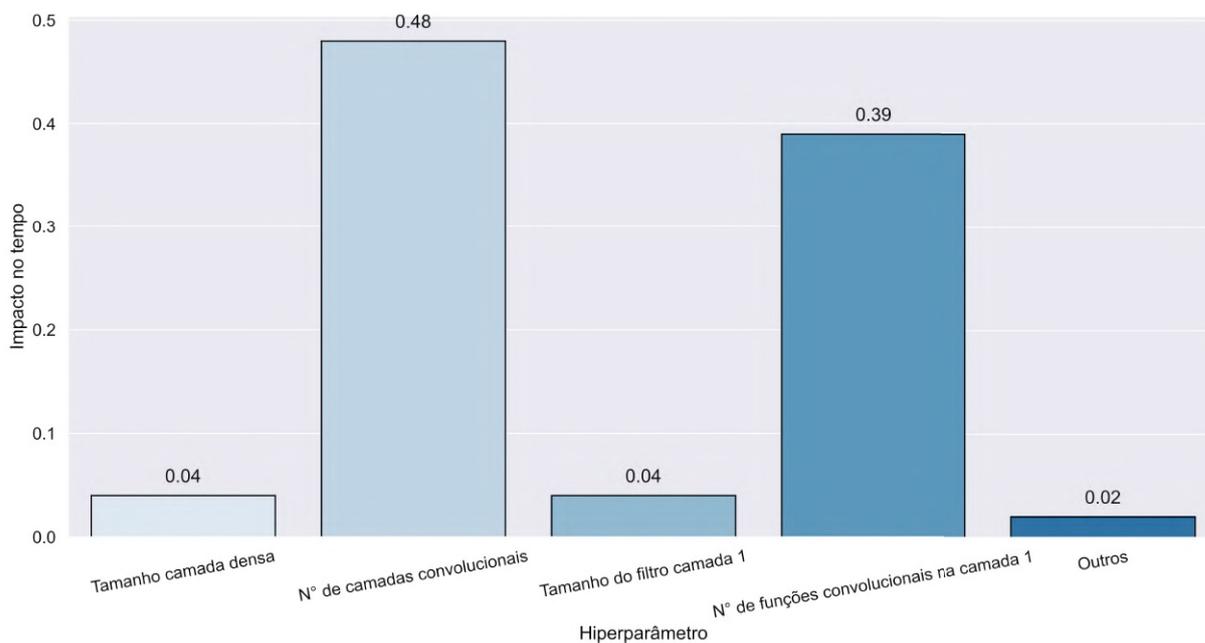
FONTE: Próprio autor

LEGENDA: Histórico de otimização do modelo CNN, em que cada ponto é uma configuração testada e o valor de F-1 Score obtido.

FIGURA 59 – IMPACTO RELATIVO DE CADA HIPERPARÂMETRO NO DESEMPENHO E NO TEMPO DO MODELO DO CNN



(a) importância relativa no desempenho



(b) impacto relativo no tempo de otimização

FONTE: Próprio autor

LEGENDA: Impacto relativo de cada hiperparâmetro no resultado e no tempo de otimização para o modelo de CNN.

## 7 DISCUSSÃO

Este capítulo aprofunda nos resultados e discussões apresentados no capítulo 6, com objetivo de realizar maiores comparações, elaborar hipóteses de explicações e comparar com resultados de trabalhos similares. Também são apresentadas ideias para teste futuros.

### 7.1 COMPARATIVO GERAL DOS MODELOS

Analisando os resultados individuais de cada modelo, o modelo SVM no experimento I apresentou o maior F1-score dentre os modelos de ML. Em termos de acuracidade, as maiores acuracidades com os modelos de ML foram obtidas pelos modelos AdaBoost no experimento I e o modelo SVM no experimento II. Quando comparados aos modelos de ML e DL, os modelos de DL ficaram, na maioria das comparações, com valores de F1-score menor. Isso pode ser atribuído a diferentes fatores, desde das arquiteturas testadas quanto a quantidade de dados utilizadas para treinamento. Quanto ao tempo de otimização, os resultados não são comparáveis, uma vez que as máquinas utilizadas são distintas e no caso dos modelos de DL, são apropriadas para executar modelos DL.

Nos modelos de ML, os resultados de uma forma geral foram similares. Alguns modelos testados são mais complexos que outros, com um maior número de hiperparâmetros e utilizando estratégias de *ensemble*, o que aumenta muito o tempo necessário para treinamento. Para comparação o modelo de SVM no experimento I levou quase 13 minutos para otimização, enquanto o modelo *bagging*, por exemplo, no experimento com UMAP, que reduziu o espaço de características, levou cerca de 39 minutos ou cerca de três vezes mais tempo. Isso mostra que para o problema de classificação de eventos de AOS, a complexidade do modelo não é, necessariamente, um fator para melhores resultados, de modo que ainda existe muitos tópicos e hipóteses a serem estudadas com modelos tidos como mais simples.

### 7.2 COMPARATIVO DOS EXPERIMENTOS DE DIMENSIONALIDADE

A TABELA 60 apresenta um comparativo com as médias dos resultados obtidos por cada modelo de ML em cada um dos experimentos. Na TABELA, é destacado o melhor valor obtido de cada resultado. Os melhores resultados de acuracidade e *recall*, número de verdadeiros positivos e falsos positivos foram obtidos no experimento sem redução de dimensionalidade, enquanto que o melhor valor de sensibilidade, F1-score, verdadeiros negativos e falsos negativos foram obtidos pelo experimento com UMAP.

Embora o experimento com UMAP tenha melhor classificado eventos de não-AOS, também foi o com maior número de falsos positivos e o menor número de verdadeiros positivos. A redução de dimensionalidade aplicada pelo PCA também aumentou o número de eventos de não-AOS classificados corretamente. O experimento PCA, no geral, apresentou bons resultados, próximos dos melhores obtidos. Isso pode indicar que a redução de dimensionalidade pode ajudar os modelos no reconhecimento dos padrões da AOS em sinais de ECG, em especial na identificação de eventos não-AOS, com a vantagem do menor tempo de modelagem e treinamento (com uma redução de quase 40% no tempo do experimento I para o II). Como os melhores resultados para predição de eventos de AOS foram obtidos no experimento de DL, é possível que exista um padrão para AOS que não foi totalmente bem capturado pelos modelos de ML e que podem ser testados com outras configurações e até outras técnicas de redução de dimensionalidade. Estratégias mistas, em que modelos com as 23 características tem peso maior na predição e de eventos de AOS enquanto que modelos com um espaço reduzido de características podem ter peso maior na detecção de eventos de não-AOS pode gerar bons resultados.

TABELA 60 – COMPARATIVO ENTRE OS EXPERIMENTOS COM AS MÉDIAS DOS RESULTADOS OBTIDOS

Experimento	Acuracidade	Recall	Precisão	F1-score	TP	FP	FN	TN
I - Sem redução	<b>0,746</b>	<b>0,792</b>	0,804	0,795	4246	<b>2280</b>	2089	8576
II - PCA	0,742	0,777	0,820	0,797	4013	2513	1923	8742
III - t-SNE	0,725	0,762	0,811	0,786	3818	2708	2012	8653
IV - UMAP	0,740	0,763	<b>0,844</b>	<b>0,801</b>	3722	2804	<b>1669</b>	<b>8996</b>
DL	0,719	0,742	0,781	0,778	<b>4964</b>	3423	1884	6914

FONTE: Elaborada pelo autor

### 7.3 COMPARATIVO COM TRABALHOS SIMILARES

A TABELA 61 apresenta um comparativo dos resultados com trabalhos similares encontrados na literatura. Na TABELA, são apresentados os melhores resultados de cada trabalho, uma vez que vários autores treinaram diferentes modelos para fins de comparação. Também foram listados os trabalhos que utilizaram a mesma base de dados. Os modelos aqui construídos tiveram um desempenho, de forma geral, pior em acuracidade mas semelhantes em *F1-score*. A coluna "divisão PhysioNet" aponta se o trabalho seguiu ou não seguindo a recomendação da PhysioNet (conforme explicado no capítulo 5). É possível notar que utilização da recomendação impactou os resultados. Alguns trabalhos obtiveram altos índices de acuracidade, acima de 90%, utilizando a recomendação, como Yang et al. (2022) e Lin et al. (2022), mas os demais obtiveram uma acuracidade em torno de 80%. Enquanto os trabalhos que não seguiram, obtiveram resultados entre 85 e 90%. A menor acuracidade obtida nesta dissertação pode

ter acontecido pelo fato de ter utilizado menos dados, seguindo a recomendação da PhysioNet, como também em algumas das diferenças nas metodologias aplicadas, conforme explorado no capítulo 4. Outra possibilidade é de que a otimização visou maximizar o F1-score, para obter um melhor equilíbrio de verdadeiros positivos e falsos negativos.

TABELA 61 – COMPARATIVO DE RESULTADOS COM TRABALHOS SIMILARES DA LITERATURA

Trabalho	Modelo	Acurácia (%)	Recall (%)	Precisão (%)	F1-score	Divisão PhysioNet
Yang et al. (2022)	Redes residuais (DL)	90,30	84,70	-	0,873	Sim
Lin et al. (2022)	Ensemble de modelos	91,40	89,80	-	-	Não
Bahrami e Forouzanfar (2022)	DL híbrido CNN e LSTM	88,13	81,49	-	0,840	Não
Almutairi et al. (2021)	DL híbrido CNN e LSTM	90,92	91,24	-	0,928	Sim
Faust et al. (2021)	LSTM	81,30	59,90	78,00*	0,678*	Sim
Mukherjee et al. (2021)	Ensemble de DL	85,58	84,43	84,80	0,848	Não
Feng et al. (2021)	Tempo-dependente custo-sensitivo (DL)	80,70	80,10	72,40	0,760	Não
Sharan et al. (2021)	Redes residuais (DL)	89,30	85,62	-	-	Sim
Fatimah et al. (2020)	SVM	92,59	94,67	91,27	0,929*	Não
Padovano et al. (2020)	SVM	81,40	82,00	-	-	Sim
Sadr e Chazal (2019)	LDA	84,40	82,90	-	-	Sim
Pinho et al. (2019)	MLP	82,12	88,41	-	-	Não
Li et al. (2018a)	Autoencoder com MLP	83,80	88,90	-	-	Sim
Sadr et al. (2015)	ELM	82,50	81,90	-	-	Sim
Este trabalho	SVM (Experimento I)	77,00	80,00	83,90	0,819	
	SVM (Experimento II)	77,20	80,00	84,30	0,821	Sim
	SVM (Experimento III)	74,20	76,20	84,90	0,803	
	SVM (Experimento IV)	75,00	75,90	87,50	0,813	

FONTE: Elaborada pelo autor

NOTA: \* Calculados pelo autor com os dados fornecidos no artigo

Analisando a TABELA, o modelo de SVM foi listado como melhor em diferentes trabalhos, inclusive em todos os experimentos de ML realizados neste trabalho. Alguns modelos de DL apresentaram acurácia elevada, mas um F1-score baixo, indicando uma dificuldade para classificar uma classe corretamente. Isso demonstra que existem ainda muitas lacunas a serem exploradas com os modelos tradicionais, que estão desempenhando bem, e com os modelos de DL. Os modelos de DL encontrados na literatura seguem arquiteturas bem definidas, como por exemplo no trabalho de Bahrami e Forouzanfar (2022), além de uma maior quantidade de dados para treinamento, conforme já foi discutido. Estas arquiteturas definidas ajudaram na acuracidade dos modelos de DL e podem ser testadas junto com Optuna para melhor calibrar os hiperparâmetros ou adicionar um pouco de customização, com a metodologia deste trabalho. Também existem lacunas a serem estudadas com modelos de DL na classificação de eventos de AOS.

Nos trabalhos estudados, a seleção de hiperparâmetros se deu predominantemente de duas de formas diferentes. Ou os hiperparâmetros foram fixados, ou foram selecionados de forma empírica, em que algumas configurações foram testadas com base de treinamento e validação. Não foram realizados estudos de impactos de desempenho ou tempo de treinamento de cada hiperparâmetro. Neste aspecto, as discussões aqui levantadas mostram que avaliar bem os hiperparâmetros ajudam na hora criar novas hipóteses, testes e no entendimento do problema.

Alguns trabalhos empregaram técnicas de redução de dimensionalidade. Bahrami e Forouzanfar (2022) aplicaram PCA em um espaço de características de 27 características. Os autores não fizeram experimento com e sem aplicação do PCA e também não comentaram quantas componentes utilizaram. Buscando entender mais sobre os impactos das características na classificação, os autores treinaram uma DT para este fim, medindo o valor do peso de cada característica. Encontraram a HF e VLF como as principais características para detectar a apneia do sono. Sadr e Chazal (2019) também utilizaram PCA, realizando experimentos com e sem aplicação do PCA. O melhor resultado encontrado foi com o uso do PCA.

Outros trabalhos aplicaram estratégias de seleção de características, que também diminuem o espaço de características, mas diferentemente das técnicas de redução aqui aplicadas, que transformam o espaço de características em novas características, a seleção apenas elimina características consideradas menos relevantes para o problema de classificação. No trabalho de Fatimah et al. (2020), os autores aplicaram o teste estatístico de hipóteses não-paramétrico Kruskal-Wallis em sua base de características constituída de entropias calculadas (um tipo de característica extraídas das séries R-R que não foram avaliadas neste trabalho) para selecionar as melhores características, assumindo que as amostras das características possuem a mesma

distribuição contínua, removendo as características com p-valor menor que 0,05.

Padovano et. al. (PADOVANO et al., 2020) aplicaram o teste de hipóteses de Welch para determinar as características mais importantes além de aplicar experimentos com diferentes subconjuntos de características para treinamento (com 19, 15 e 14 características). encontrando VLF, LF, HF, RMSSD, pNN50, SDSD, SDNN, entre outras, como as que mais ajudaram na classificação. Os experimentos mostraram que subconjuntos com maior ou menor número de características alcançaram resultados similares.

O método de discriminante de relevância foi utilizada no trabalho de Pinho et al. (2019), para selecionar as características. Os autores também realizaram diversos experimentos variando a quantidade de características disponíveis para treinamento (que variaram em combinações de 84 a 6 características). Em termos de assertividade, o melhor resultado encontrado foi com um subconjunto de 20 características.

Ivanko et al. (2020) também aplicaram experimentos com um conjunto de 24 e 9 características. Os modelos testados obtiveram resultados elevados de acuracidade em ambos os casos.

Estes achados estão em sincronia com os resultados aqui obtidos. A análise de explicabilidade, que buscou avaliar o impacto de cada características após a decisão do modelo, mostrou que as características do domínio da frequência, em especial a VLF, tiveram maior impacto. Características do domínio do tempo, como SDNN-RR, também tiveram relevância neste processo. Os experimentos aqui realizados mostraram que a redução de dimensionalidade pode ajudar, mas as 23 características originais utilizadas ainda possuem informações que a redução de dimensionalidade não conseguiu capturar. Conforme visto em outros trabalhos, o PCA se mostrou um método promissor e com bons resultados na classificação da AOS.

#### 7.4 RESUMO

Neste capítulo foi aprofundada a discussão dos resultados iniciadas no capítulo 6. Os resultados encontrados foram comparados com outros trabalhos, mostrando que no contexto de classificação de AOS, ainda existe muito avanço para ser feito na seleção dos hiperparâmetros e explicabilidade dos modelos.

## **Parte IV**

### **Conclusão**

## 8 CONCLUSÃO

A AOS acomete uma grande quantidade de pacientes que, muitas vezes, não são corretamente diagnosticadas, prejudicando substancialmente a qualidade de vida destas pessoas. O auxílio diagnóstico utilizando o sinal não evasivo do do ECG é possível e pode ser mais uma ferramenta à disposição da equipe médica para decidir sobre o caso e orientar o paciente. Neste âmbito, o *framework* aqui desenvolvido foi capaz de comparar diferentes técnicas de ML e DL para classificação de eventos de AOS. Além de construir modelos computacionais capazes de detectar episódios de AOS, o *framework* desenvolvido enriqueceu a discussão trazendo dados sobre como e porquê tais modelos tomaram a decisão se em uma janela de 1 minuto de ECG o paciente estava sofrendo de AOS. Deste modo, um especialista da área médica pode avaliar as características extraídas das séries R-R que apontaram para classificação daquele episódio. Isto é extremamente importante para aumentar a transparência dos resultados, a confiabilidade do modelo e fomentar novas discussões. Embora possa ser complexo para um especialista médico avaliar estas importâncias dentro das técnicas de redução de dimensionalidade, estes dados podem ajudar nos profissionais encarregados de desenvolver, treinar e avaliar o modelo.

Dentro deste aspecto, o *framework* desenvolvido avalia diferentes configurações de hiperparâmetros que irão compor o modelo, trazendo dados sobre seu impacto nos resultados e tempo de processamento. Estes dados ajudam os desenvolvedores encarregados a melhor entender o problema, detectar possíveis casos de *overfitting* e *underfitting*. Os dados também ajudam a ponderar sobre quais modelos utilizar e como um fluxo de processamento de dados pode ser impactado pelos hiperparâmetros. Por exemplo, em uma aplicação que utiliza um modelo treinado, os dados fornecidos trazem os tempos de predição para centenas janelas de ECG. Isto ajuda os engenheiros da aplicação melhor prepará-la ao entender a taxa de resposta do modelo. Caso o modelo necessite de ser retreinado, o impacto de cada hiperparâmetro foi medido e os engenheiros podem decidir por quais hiperparâmetros realizarem uma nova sintonia e quais manter com os valores já encontrados.

### 8.1 OBJETIVOS DA DISSERTAÇÃO

O objetivo geral e os objetivos específicos desta dissertação foram descritos no 1. Com base nos resultados atingidos, é possível analisar e discutir cada objetivo proposto. Para os objetivos específicos:

- Uma pesquisa bibliográfica sobre classificação de AOS utilizando técnicas de ML

e DL foi realizada e apresentada no capítulo 4, discutindo a metodologia utilizada, os eixos de pesquisa e apresentando mapas visuais das palavras-chaves. Este objetivo foi totalmente concluído;

- Dentro do objetivo de aplicar algoritmos para extração das características a partir dos sinais de ECG, o *framework* implementado em Python atendeu completamente o objetivo, sendo capaz de processar os sinais de ECG da base de dados pública PhysioNet, possibilitando tanto a extração de séries R-R e séries de pico R quanto a extração de 23 diferentes características das séries R-R. O *framework* ainda aplica o filtro proposto por Chen e Guestrin (2016) nos sinais de ECG;
- Outro objetivo concluído foi a avaliação das técnicas de redução da dimensionalidade e estudo das características, pois o *framework* é capaz de aplicar ou não técnicas de redução de dimensionalidade. Até o presente momento, o *framework* é capaz de aplicar as todas as técnicas listadas no objetivo: PCA, t-SNE e UMAP. O *framework* foi construído de tal modo que no futuro adicionar outras técnicas, tais como a análise de fatores, por exemplo. Além de aplicar as técnicas de redução de dimensionalidade, o *framework* fornece dados e gráficos para comparação das características na importância e impacto no resultado de cada modelo;
- O *framework* atendeu totalmente ao objetivo de realizar uma busca Bayesiana, pois dentro do fluxo de treinamento e teste de modelos de ML e DL, a busca de hiperparâmetros é realizada com uso da ferramenta Optuna, que realiza uma otimização Bayesiana, tanto para os modelos de ML quanto para os modelos de DL;
- Para o objetivo de classificar as características extraídas e os sinais processados de ECG quanto a ocorrência ou não de eventos de apneia do sono utilizando modelos de ML e DL, existe uma leve diferença para que foi estabelecido no objetivo para a implementação do *framework*, pois os modelos de ML foram treinados exclusivamente com as características extraídas enquanto os modelos de DL foram treinados exclusivamente com os sinais interpolados das séries R-R e de pico R. Isso se deu por conta do tipo de dado que é imputado nos modelos de ML e DL serem diferentes. Fora esta pequena diferença, ambos os modelos foram capazes de classificar eventos de AOS. Apesar desta ligeira diferença, pode-se considerar o objetivo como concluído.
- O objetivo de analisar os resultados obtidos dos modelos preditivos utilizando técnicas de desempenho de classificação foi concluído com *framework*, em que o desempenho de cada modelo é calculado utilizando todas as métricas estabelecidas para os resultados de teste. Adicionalmente às métricas estabelecidas no

objetivo, o *framework* é capaz realizar dois tipos diferentes de teste de hipóteses para os modelos de DL e ML;

- Os valores SHAP também foram calculados e analisados para maioria dos modelos de ML implementados. Este objetivo foi parcialmente concluído, pois por limitações computacionais não foi possível calcular os valores SHAP para os modelos VM, SVM e *bagging*;
- A comparação e discussão dos resultados obtidos com artigos de classificação de AOS utilizando técnicas de ML e DL encontrados na literatura recente é apresentada no capítulo 6, concluindo o objetivo.

Por fim, o objetivo principal da dissertação foi concluído. O *framework* é capaz de realizar todas as tarefas designadas, sendo capaz de classificar sinais de ECG de canal único na ocorrência ou não de eventos de AOS. O *framework* ainda foi testado com uma base de dados pública. Existem melhorias que podem ser aplicadas e possibilidades de novos trabalhos. Estas melhorias e recomendações de trabalhos futuros serão discutidos na próxima seção.

## 8.2 PERSPECTIVAS DE FUTURAS PESQUISA

Os resultados de classificação com os diferentes modelos treinados, além dos resultados de explicabilidade e de otimização mostram que o problema de identificação da AOS ainda pode ser mais explorado. Para trabalhos futuros é interessante aplicar o *framework* aqui desenvolvido em outras bases de dados, tanto públicas disponíveis na Internet quanto bases coletadas por hospitais e centros de pesquisa. A redução de dimensionalidade se mostrou importante para obter bons resultados, nesse aspecto, incluir outras técnicas de redução de dimensionalidade, como IPCA (PCA incremental, do inglês *Incremental Principal Component Analysis*), entre outras, podem trazer resultados interessantes. Adicionar outras formas de processar o ECG, como a transformada de *wavelet* pode ajudar na comparação de diferentes características na detecção de eventos de AOS.

Dentro dos experimentos com DL, adicionar métodos de trazer a explicabilidade dos modelos é um grande incremento ao *framework* desenvolvido. O trabalho de Anand et al. (2022a) propõe uma modelo de DL explicável com uso de valores SHAP. Esta metodologia poderia ser incluída no *framework*. Arquiteturas prontas de modelos de DL também podem ser aplicadas, utilizando o *framework* para comparações dos resultados, uma vez que a estrutura das redes dessas arquiteturas é definida. Para os modelos de ML, outros tipos de modelos podem ser testados, como por exemplo KNN, e uma amplitude maior de hiperparâmetros podem ser configurados. Além de explorar outros

tipos de modelo, é também possível estudar metamodelos, que combinam e agrupam os modelos já desenvolvidos e estudados com técnicas de *ensemble*.

Dentro da classificação de eventos de AOS, o *framework* poderia ser ampliado para processar outros tipos de sinais fisiológicos, tais como EEG, variações na pulsação, fluxo de oxigênio, entre outros. Além de explorar a classificação de eventos de AOS, o *framework* poderia ser adaptado para classificar distorções do ECG de outras cardiopatias.

Outras linhas de pesquisa futura envolvem a geração de dados sintéticos, tanto para balancear as classes, quanto para gerar novos exemplos de dados de ECG de eventos de AOS e não-AOS. Embora a base trabalhada nesta dissertação seja razoavelmente bem equilibrada e com cerca de 34 mil pontos de dados, os modelos de ML e principalmente DL podem se beneficiar de uma base de dados maior e totalmente equilibrada entre as classes.

## REFERÊNCIAS

- ABADI, M.; AGARWAL, A.; BARHAM, P.; BREVDO, E.; CHEN, Z.; CITRO, C.; CORRADO, G.; DAVIS, A.; DEAN, J.; DEVIN, M.; GHEMAWAT, S.; GOODFELLOW, I.; HARP, A.; IRVING, G.; ISARD, M.; JIA, Y.; JOZEFOWICZ, R.; KAISER, L.; KUDLUR, M.; LEVENBERG, J.; MANÉ, D.; MONGA, R.; MOORE, S.; MURRAY, D.; OLAH, C.; SCHUSTER, M.; SHLENS, J.; STEINER, B.; SUTSKEVER, I.; TALWAR, K.; TUCKER, P.; VANHOUCHE, V.; VASUDEVAN, V.; VIÉGAS, F.; VINYALS, O.; WARDEN, P.; WATTENBERG, M.; WICKE, M.; YU, Y.; ZHENG, X. **TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems**. [S.l.: s.n.], 2015. <https://www.tensorflow.org/>. Software aberto. Versão 2.0. Acessado em 2024-02-02. Citado na p. 136.
- ABDOLRASOL, M. G. M.; HUSSAIN, S. M. S.; USTUN, T. S.; SARKER, M. R.; HANNAN, M. A.; MOHAMED, R.; ALI, J. A.; MEKHILEF, S.; MILAD, A. Artificial Neural Networks Based Optimization Techniques: A Review. **Electronics**, v. 10, n. 21, 2021. Citado nas pp. 61–63.
- ABI-RAMIA, L. B. P.; CARVALHO, F. A. R.; COSCARELLI, C. T.; ALMEIDA, M. A. d. O. Aparelho de avanço mandibular aumenta o volume da via aérea superior de pacientes com apneia do sono. **Dental Press Journal of Orthodontics**, v. 15, p. 166–171, 2010. Citado na p. 42.
- ABIODUN, O. I.; JANTAN, A.; OMOLARA, A. E.; DADA, K. V.; MOHAMED, N. A.; ARSHAD, H. State-of-the-art in artificial neural network applications: A survey. **Heliyon**, v. 4, n. 11, 2018. Citado nas pp. 59, 60, 62.
- AFIFAH, K.; YULITA, I. N.; SARATHAN, I. Sentiment Analysis on Telemedicine App Reviews using XGBoost Classifier. In: IEEE. 2021 International Conference on Artificial Intelligence and Big Data Analytics. Bandungue, Indonésia: [s.n.], 2021. P. 22–27. Citado na p. 71.
- AGGARWAL, K.; KHADANGA, S.; JOTY, S.; KAZAGLIS, L.; SRIVASTAVA, J. A structured learning approach with neural conditional random fields for sleep staging. In: IEEE. 2018 IEEE International Conference on Big Data (Big Data). Seattle, Estados Unidos: [s.n.], 2018. P. 1318–1327. Citado na p. 106.

AKBAR, W.; WU, W.-p.; SALEEM, S.; FARHAN, M.; SALEEM, M. A.; JAVEED, A.; ALI, L. Development of hepatitis disease detection system by exploiting sparsity in linear support vector machine to improve strength of adaboost ensemble model. **Mobile Information Systems**, v. 2020, p. 1–9, 2020. Citado na p. 68.

AKIBA, T.; SANO, S.; YANASE, T.; OHTA, T.; KOYAMA, M. Optuna: A Next-generation Hyperparameter Optimization Framework. In: PROCEEDINGS of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Anchorage, Estados Unidos: Association for Computing Machinery, 2019. Citado nas pp. 82–84, 129, 142.

ALMUTAIRI, H.; HASSAN, G. M.; DATTA, A. Classification of Obstructive Sleep Apnoea from single-lead ECG signals using convolutional neural and Long Short Term Memory networks. **Biomedical Signal Processing and Control**, v. 69, ago. 2021. Citado nas pp. 101, 102, 107, 115, 121, 177.

ALSAGHEER, R.; ALHARAN, A. F.; AL-HABOOBI, A. Popular decision tree algorithms of data mining techniques: a review. **International Journal of Computer Science and Mobile Computing**, v. 6, n. 6, p. 133–142, 2017. Citado na p. 65.

ALZUBAIDI, L.; ZHANG, J.; HUMAIDI, A. J.; AL-DUJAILI, A.; DUAN, Y.; AL-SHAMMA, O.; SANTAMARÍA, J.; FADHEL, M. A.; AL-AMIDIE, M.; FARHAN, L. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. **Journal of big Data**, v. 8, p. 1–74, 2021. Citado nas pp. 90–92.

ANAND, A.; KADIAN, T.; KUMAR, M. S.; GUPTA, A. Explainable AI decision model for ECG data of cardiac disorders. **Biomedical Signal Processing and Control**, v. 75, p. 103584, 2022. ISSN 1746-8094. Citado na p. 183.

ANAND, A.; KADIAN, T.; SHETTY, M. K.; GUPTA, A. Explainable AI decision model for ECG data of cardiac disorders. **Biomedical Signal Processing and Control**, Elsevier, v. 75, p. 103584, 2022. Citado na p. 88.

ANÔNIMO. **LightGBM - Python-package Introduction**. [S.l.: s.n.]. <https://lightgbm.readthedocs.io/en/latest/Python-Intro.html>. Software livre disponibilizado pela Microsoft. Versão 4.3.0. Acessado em 2024-02-02. Citado na p. 130.

ANÔNIMO. **O que é o CPAP?** [S.l.]: Instituto do Sono.  
<https://institutodosono.com/artigos-noticias/cpap-o-que-e/>. Acessado em 2024-02-02.  
Citado na p. 42.

\_\_\_\_\_. **XGBoost - XGBoost Documentation.** [S.l.: s.n.].  
<https://xgboost.readthedocs.io/en/latest/index.html>. Software aberto. Versão 2.0.  
Acessado em 2024-02-02. Citado na p. 130.

ARAGHI, M. H.; CHEN, Y. F.; JAGIELSKI, A.; CHOUDHURY, S.; BANERJEE, D.; HUSSAIN, S.; THOMAS, G. N.; TAHERI, S. Effectiveness of lifestyle interventions on obstructive sleep apnea (OSA): Systematic review and meta-analysis. **Sleep**, v. 36, p. 1553–1562, 10 out. 2013. Citado na p. 40.

ATALLAH, R.; AL-MOUSA, A. Heart disease detection using machine learning majority voting ensemble method. In: 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS). Amã, Jordânia: [s.n.], 2019. P. 1–6. Citado na p. 73.

AYASHM, S.; CHEHEL AMIRANI, M.; VALIZADEH, M. Analysis of ecg signal by using an fcn network for automatic diagnosis of obstructive sleep apnea. **Circuits, Systems, and Signal Processing**, Springer, v. 41, n. 11, p. 6411–6426, 2022. Citado na p. 109.

AYATOLLAHI, A.; AFRAKHTEH, S.; SOLTANI, F.; SALEH, E. Sleep apnea detection from ECG signal using deep CNN-based structures. **Evolving Systems**, Springer, v. 14, n. 2, p. 191–206, 2022. Citado na p. 109.

AYONARA, D.; SILVA, M. H. L. D.; VASCONCELOS, C. C. D.; COSTA, R. D. O.; RIBEIRO, R.; COSTA, S. Set-Out. **Revista CEFAC**, v. 16, p. 1621–1626, 5 2014.  
Citado nas pp. 29, 34, 39–42, 126.

BABY, B. S.; ARONOW, W. S.; CHANDY, D. Therapeutic options for obstructive sleep apnea. **American Journal of Therapeutics**, v. 20, n. 5, p. 534–542, 2013. Citado nas pp. 40, 42.

BAHRAMI, M.; FOROUZANFAR, M. Sleep apnea detection from single-lead ECG: a comprehensive analysis of machine learning and deep learning algorithms. **IEEE Transactions on Instrumentation and Measurement**, v. 71, 2022. Citado nas pp. 29, 30, 99, 101, 109, 115, 120, 125, 126, 177, 178.

BALCI, M.; TASDEMIR, S.; OZMEN, G.; GOLCUK, A. Machine learning-based detection of sleep-disordered breathing type using time and time-frequency features. **Biomedical Signal Processing and Control**, v. 73, mar. 2022. Citado nas pp. 97, 99, 101, 108.

BAYINDIR, R.; YESILBUDAK, M.; COLAK, M.; GENÇ, N. A novel application of naive bayes classifier in photovoltaic energy prediction. In: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA). Cancún, México: [s.n.], 2017. P. 523–527. Citado na p. 54.

BENJAFIELD, A. V.; AYAS, N. T.; EASTWOOD, P. R.; HEINZER, R.; IP, M. S.; MORRELL, M. J.; NUNEZ, C. M.; PATEL, S. R.; PENZEL, T.; PÉPIN, J. L. D.; PEPPARD, P. E.; SINHA, S.; TUFIK, S.; VALENTINE, K.; MALHOTRA, A. Estimation of the global prevalence and burden of obstructive sleep apnoea: a literature-based analysis. **The Lancet Respiratory Medicine**, v. 7, p. 687–698, 8 ago. 2019. Citado na p. 29.

BERGSTRA, J.; BARDENET, R.; BENGIO, Y.; KÉGL, B. Algorithms for hyper-parameter optimization. In: ADVANCES in Neural Information Processing Systems. Granada, Espanha: [s.n.], 2011. v. 24. Citado na p. 84.

BEWICK, V.; CHEEK, L.; BALL, J. Statistics review 14: Logistic regression. **Critical care**, v. 9, p. 1–7, 2005. Citado na p. 55.

BISCHL, B.; BINDER, M.; LANG, M.; PIELOK, T.; RICHTER, J.; COORS, S.; THOMAS, J.; ULLMANN, T.; BECKER, M.; BOULESTEIX, A.-L.; DENG, D.; LINDAUER, M. Hyperparameter optimization: foundations, algorithms, best practices, and open challenges. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, v. 13, n. 2, e1484, 2023. Citado na p. 82.

BORUCKA, A.; GRZELAK, M. Application of logistic regression for production machinery efficiency evaluation. **Applied Sciences**, v. 9, n. 22, p. 4770, 2019. Citado na p. 55.

BOZKURT, F.; UÇAR, M. K.; BILGIN, C.; ZENGİN, A. Sleep-wake stage detection with single channel ECG and hybrid machine learning model in patients with obstructive sleep apnea. **Physical and Engineering Sciences in Medicine**, v. 44, p. 63–77, 1 mar. 2021. Citado nas pp. 99, 101, 107.

BREIMAN, L. Random forests. **Machine learning**, v. 45, p. 5–32, 2001. Citado na p. 66.

CAI, Y.-D.; RICARDO, P.-W.; JEN, C.-H.; CHOU, K.-C. Application of SVM to predict membrane protein types. **Journal of Theoretical Biology**, v. 226, n. 4, p. 373–376, 2004. Citado na p. 59.

CARLILES, S.; BUDAÁRI, T.; HEINIS, S.; PRIEBE, C.; SZALAY, A. S. Random forests for photometric redshifts. **The Astrophysical Journal**, v. 712, n. 1, p. 511, 2010. Citado na p. 67.

CARMO, J.; ARAÚJO, I.; MARQUES, F.; FONSECA, C. Sleep disordered breathing in heart failure: The state of the art after the SERVE-HF trial. **Revista Portuguesa de Cardiologia**, v. 36, p. 859–867, 11 nov. 2017. Citado na p. 29.

CARVALHO, D. V.; PEREIRA, E. M.; CARDOSO, J. S. Machine learning interpretability: a survey on methods and metrics. **Electronics**, v. 8, n. 8, 2019. Citado na p. 51.

CERRI, R.; LEON FERREIRA, A. C. P. de et al. Aprendizado de máquina: breve introdução e aplicações. **Cadernos de Ciência & Tecnologia**, v. 34, n. 3, p. 297–313, 2019. Citado nas pp. 45, 48, 49.

CHAMPSEIX, R. **hrv-analysis 1.0.4 - a package to calculate features from RR Interval for HRV analyses**. [S.l.: s.n.]. <https://pypi.org/project/hrv-analysis/>. Software aberto. Versão 1.0.4. Acessado em 2024-02-02. Citado na p. 127.

CHANG, Y.-C.; CHANG, K.-H.; WU, G.-J. Application of eXtreme gradient boosting trees in the construction of credit risk assessment models for financial institutions. **Applied Soft Computing**, v. 73, p. 914–920, 2018. Citado na p. 71.

CHAUHAN, N. K.; SINGH, K. A Review on Conventional Machine Learning vs Deep Learning. In: 2018 International Conference on Computing, Power and Communication Technologies (GUCON). Grande Noida, Índia: [s.n.], 2018. P. 347–352. Citado na p. 89.

CHAZAL, P. D.; HENEGHAN, C.; SHERIDAN, E.; REILLY, R.; NOLAN, P.; O'MALLEY, M. Automated processing of the single-lead electrocardiogram for the detection of obstructive sleep apnoea. **IEEE Transactions on Biomedical Engineering**, v. 50, p. 686–696, 6 jun. 2003. Citado na p. 120.

CHEN, L.; ZHANG, X.; SONG, C. An automatic screening approach for obstructive sleep apnea diagnosis based on single-lead electrocardiogram. **IEEE transactions on automation science and engineering**, v. 12, n. 1, p. 106–115, 2014. Citado nas pp. 97, 109, 122–124.

CHEN, T.; GUESTRIN, C. XGBoost: a scalable tree boosting system. In. Citado nas pp. 70, 71, 182.

CHEN, W.-H.; HSU, S.-H.; SHEN, H.-P. Application of SVM and ANN for intrusion detection. **Computers and Operations Research**, v. 32, n. 10, p. 2617–2634, 2005. Citado na p. 59.

CHOLLET, F. et al. **Keras**. [S.l.: s.n.], 2015. <https://keras.io>. Software aberto. Versão 3.0. Acessado em 24/02/2024. Citado na p. 136.

CLIFFORD, G. **Signal processing methods for heart rate variability**. 2002. Tese (Doutorado) – Oxford University, Oxford, Reino Unido. Citado na p. 126.

DANIELS, H.; KAMP, B. Application of MLP networks to bond rating and house pricing. **Neural Computing & Applications**, v. 8, p. 226–234, 1999. Citado na p. 63.

DAS, K.; BEHERA, R. N. A survey on machine learning: concept, algorithms and applications. **International Journal of Innovative Research in Computer and Communication Engineering**, v. 5, n. 2, p. 1301–1309, 2017. Citado na p. 45.

DATTA, S.; PIHUR, V.; DATTA, S. An adaptive optimal ensemble classifier via bagging and rank aggregation with applications to high dimensional data. **BMC bioinformatics**, v. 11, n. 1, p. 1–11, 2010. Citado na p. 73.

DELASHMIT, W. H.; MANRY, M. T. et al. Recent developments in multilayer perceptron neural networks. In: PROCEEDINGS of the seventh annual memphis area engineering and science conference. Memphis, Estados Unidos: [s.n.], 2005. v. 7, p. 33. Citado na p. 60.

DEY, R.; SALEM, F. M. Gate-variants of gated recurrent unit (GRU) neural networks. In: IEEE. 2017 IEEE 60th international midwest symposium on circuits and systems. Boston, Estados Unidos: [s.n.], 2017. P. 1597–1600. Citado na p. 94.

DOGRU, N.; SUBASI, A. Traffic accident detection using random forest classifier. In: IEEE. 2018 15th learning and technology conference. Gidá, Arábia Saudita: [s.n.], 2018. P. 40–45. Citado na p. 67.

DOMASHOVA, J.; MIKHAILINA, N. Usage of machine learning methods for early detection of money laundering schemes. **Procedia Computer Science**, v. 190, p. 184–192, 2021. Citado na p. 84.

DONG, X.; YU, Z.; CAO, W.; SHI, Y.; MA, Q. A survey on ensemble learning. **Frontiers of Computer Science**, v. 14, p. 241–258, 2020. Citado na p. 71.

ERDENEBAYAR, U.; KIM, Y. J.; PARK, J. U.; JOO, E. Y.; LEE, K. J. Deep learning approaches for automatic detection of sleep apnea events from an electrocardiogram. **Computer Methods and Programs in Biomedicine**, v. 180, out. 2019. Citado nas pp. 100, 101, 106.

EUROPEAN SOCIETY OF CARDIOLOGY THE NORTH AMERICAN SOCIETY OF PACING ELECTROPHYSIOLOGY, T. F. of the. Heart rate variability: standards of measurement, physiological interpretation, and clinical use. **Circulation**, v. 93, n. 5, p. 1043–1065, 1996. Citado na p. 126.

FAN, W.; STOLFO, S. J.; ZHANG, J. The application of AdaBoost for distributed, scalable and on-line learning. In: PROCEEDINGS of the fifth Association for Computing Machinery Special Interest Group on Knowledge Discovery in Data international conference on Knowledge discovery and data mining. São Diego, Estados Unidos: [s.n.], 1999. P. 362–366. Citado na p. 68.

FATIMAH, B.; SINGH, P.; SINGHAL, A.; PACHORI, R. B. Detection of apnea events from ECG segments using Fourier decomposition method. **Biomedical Signal Processing and Control**, v. 61, ago. 2020. Citado nas pp. 99, 101, 106, 177, 178.

FAUST, O.; BARIKA, R.; SHENFIELD, A.; CIACCIO, E. J.; ACHARYA, U. R. Accurate detection of sleep apnea with long short-term memory network based on RR interval signals. **Knowledge-Based Systems**, v. 212, jan. 2021. Citado nas pp. 29, 99, 101, 107, 177.

FAWAGREH, K.; GABER, M. M.; ELYAN, E. Random forests: from early developments to recent advancements. **Systems Science & Control Engineering: An Open Access Journal**, v. 2, n. 1, p. 602–609, 2014. Citado na p. 67.

FENG, K.; QIN, H.; WU, S.; PAN, W.; LIU, G. A Sleep Apnea Detection Method Based on Unsupervised Feature Learning and Single-Lead Electrocardiogram. **IEEE Transactions on Instrumentation and Measurement**, v. 70, 2021. Citado nas pp. 99, 101, 102, 107, 120, 177.

FERREIRA, A. J.; FIGUEIREDO, M. A. Boosting algorithms: A review of methods, theory, and applications. **Ensemble machine learning: Methods and applications**, p. 35–85, 2012. Citado na p. 68.

FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. Barcelona, Espanha, p. 23–37, 1995. Citado na p. 67.

FRIEDMAN, J. Stochastic gradient boosting. **Computational statistics & data analysis**, v. 38, n. 4, p. 367–378, 2002. Citado na p. 68.

FRIEDMAN, J. H. Greedy function approximation: a gradient boosting machine. **Annals of statistics**, p. 1189–1232, 2001. Citado na p. 68.

GARCIA-VICENTE, C.; GUTIÉRREZ-TOBAL, G. C.; JIMÉNEZ-GARCÍA, J.; MARTÍN-MONTERO, A.; GOZAL, D.; HORNERO, R. ECG-based convolutional neural network in pediatric obstructive sleep apnea diagnosis. **Computers in Biology and Medicine**, Elsevier, v. 167, p. 107628, 2023. Citado na p. 110.

GENT, P. van. **HeartPy - Python Heart Rate Analysis Toolkit's**. [S.l.: s.n.]. <https://python-heart-rate-analysis-toolkit.readthedocs.io/en/latest/index.html>. Software aberto. Versão 0.8.1. Acessado em 2024-07-27. Citado na p. 122.

GÉRON, A. **Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow**. 2. ed. Sebastapol, Estados Unidos: O'Reilly Media, 2019. Citado nas pp. 44, 47, 50, 56, 60–63, 67, 70, 74, 89, 92–94, 128, 130.

GLOROT, X.; BENGIO, Y. Understanding the difficulty of training deep feedforward neural networks. In: PROCEEDINGS of the thirteenth international conference on artificial intelligence and statistics. [S.l.: s.n.], 2010. P. 249–256. Citado na p. 89.

GOLDBERGER, A. L.; AMARAL, L. A.; GLASS, L.; HAUSDORFF, J. M.; IVANOV, P. C.; MARK, R. G.; MIETUS, J. E.; MOODY, G. B.; PENG, C.-K.; STANLEY, H. E. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for

complex physiologic signals. **circulation**, v. 101, n. 23, e215–e220, 2000. Citado na p. 102.

GUPTA, J.; PATHAK, S.; KUMAR, G. Deep learning (CNN) and transfer learning: A review. In: 1. JOURNAL of Physics: Conference Series. Online: [s.n.], 2022. v. 2273, p. 012029. Citado nas pp. 90, 92.

GUYTON, A. C.; HALL, J. E. **Tratado de fisiologia médica**. 11. ed. Rio de Janeiro, Brasil: Rio de Janeiro, RJ: Elsevier Editora, 2006. Citado nas pp. 34, 35.

HAMILTON, P. Open source ECG analysis. In: COMPUTERS in cardiology. Memphis, Estados Unidos: [s.n.], 2002. P. 101–104. Citado nas pp. 97, 122.

HAN, H.; OH, J. Application of various machine learning techniques to predict obstructive sleep apnea syndrome severity. **Scientific Reports**, v. 13, n. 1, p. 6379, 2023. Citado na p. 84.

HANSEN, N.; OSTERMEIER, A. Completely derandomized self-adaptation in evolution strategies. **Evolutionary computation**, v. 9, n. 2, p. 159–195, 2001. Citado na p. 84.

HART, S. Shapley value. In: GAME theory. Londres, Reino Unido: Palgrave Macmillan United Kingdom, 1989. P. 210–216. Citado nas pp. 85, 86, 88.

HASTIE, T.; ROSSET, S.; ZHU, J.; ZOU, H. Multi-class adaboost. **Statistics and its Interface**, v. 2, n. 3, p. 349–360, 2009. Citado na p. 130.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. H.; FRIEDMAN, J. H. **The elements of statistical learning: data mining, inference, and prediction**. 2. ed. Nova Iorque, Estados Unidos: Springer, 2009. Citado nas pp. 45, 50–52, 57, 58, 64, 65, 67, 70, 75, 133.

HINTON, G. E.; ROWEIS, S. Stochastic neighbor embedding. **Advances in neural information processing systems**, v. 15, 2002. Citado na p. 76.

HLAVCHEVA, D.; YALOVEHA, V.; PODOROZHNIAK, A. Application of convolutional neural network for histopathological analysis. **Advanced Information Systems**, v. 3, n. 4, p. 69–73, 2019. Citado na p. 92.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, v. 9, n. 8, p. 1735–1780, 1997. Citado na p. 93.

HOSSIN, M.; SULAIMAN, M. N. A review on evaluation metrics for data classification evaluations. **International journal of data mining & knowledge management process**, v. 5, n. 2, p. 1, 2015. Citado nas pp. 51, 52.

HUNTER, J. D. **Matplotlib**. [S.l.: s.n.]. <https://matplotlib.org/stable/>. Software aberto. Versão 3.9.1. Acessado em 2024-07-27. Citado na p. 142.

IVANKO, K.; IVANUSHKINA, N.; RYKHALSKA, A. Identifying episodes of sleep apnea in ECG by machine learning methods. In: p. 588–593. Citado nas pp. 100, 101, 106, 179.

JAMIESON, K.; TALWALKAR, A. Non-stochastic best arm identification and hyperparameter optimization. In: ARTIFICIAL intelligence and statistics. Cádiz, Espanha: [s.n.], 2016. P. 240–248. Citado na p. 83.

JAVAHERI, S.; BARBE, F.; CAMPOS-RODRIGUEZ, F.; DEMPSEY, J. A.; KHAYAT, R.; JAVAHERI, S.; MALHOTRA, A.; MARTINEZ-GARCIA, M. A.; MEHRA, R.; PACK, A. I.; POLOTSKY, V. Y.; REDLINE, S.; SOMERS, V. K. Sleep Apnea: Types, Mechanisms, and Clinical Cardiovascular Consequences. v. 69, p. 841–858, 7 fev. 2017. Citado nas pp. 29, 39, 40.

JIN, R.; YAN, F.; ZHU, J. Application of logistic regression model in an epidemiological study. **Science Journal of Applied Mathematics and Statistics**, v. 3, n. 5, p. 225–229, 2015. Citado na p. 55.

JING, W.; QIAN, B.; YANNIAN, L. Study on food safety risk based on LightGBM model: a review. **Food Science and Technology**, v. 42, 2022. Citado na p. 71.

JORDAN, M. I.; MITCHELL, T. M. Machine learning: Trends, perspectives, and prospects. **Science**, v. 349, n. 6245, p. 255–260, 2015. Citado nas pp. 45, 46, 48.

KAUR, G.; OBERAI, E. N. A review article on Naive Bayes classifier with various smoothing techniques. **International Journal of Computer Science and Mobile Computing**, v. 3, n. 10, p. 864–868, 2014. Citado nas pp. 53, 54.

KAUR, P.; KUMAR, R.; KUMAR, M. A healthcare monitoring system using random forest and internet of things (IoT). **Multimedia Tools and Applications**, v. 78, p. 19905–19916, 2019. Citado na p. 67.

KE, G.; MENG, Q.; FINLEY, T.; WANG, T.; CHEN, W.; MA, W.; YE, Q.; LIU, T.-Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In: **ADVANCES in Neural Information Processing Systems**. Long Beach, Estados Unidos: [s.n.], 2017. v. 30. Citado nas pp. 70, 71.

KULKARNI, V. Y.; SINHA, P. K. Random forest classifiers: a survey and future research directions. v. 36, n. 1, p. 1144–1153, 2013. Citado nas pp. 66, 67.

LAKSHMANAN, V.; ROBINSON, S.; MUNN, M. **Machine learning design patterns**. 1. ed. Sebastapol, Estados Unidos: O' Reilly Media, 2020. Citado nas pp. 48–50, 89.

LAVALLEY, M. P. Logistic regression. **Circulation**, v. 117, n. 18, p. 2395–2399, 2008. Citado nas pp. 54, 55.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998. Citado na p. 89.

LEI, Y.; FERRER, L.; LAWSON, A.; MCLAREN, M.; SCHEFFER, N. Application of convolutional neural networks to language identification in noisy conditions. In: **ODYSSEY 2014: The Speaker and Language Recognition Workshop**. Joensuu, Finlândia: [s.n.], 2014. Citado na p. 92.

LI, K.; PAN, W.; LI, Y.; JIANG, Q.; LIU, G. A method to detect sleep apnea based on deep neural network and hidden Markov model using single-lead ECG signal. **Neurocomputing**, v. 294, p. 94–101, jun. 2018. Citado nas pp. 30, 100, 101, 106, 120, 177.

LI, L.; JAMIESON, K.; ROSTAMIZADEH, A.; GONINA, E.; HARDT, M.; RECHT, B.; TALWALKAR, A. Massively parallel hyperparameter tuning. In: **6TH International Conference on Learning Representations 2018**. Vancouver, Canadá: [s.n.], 2018. Citado na p. 83.

LIN, C. Y.; WANG, Y. W.; SETIAWAN, F.; TRANG, N. T. H.; LIN, C. W. Sleep apnea classification algorithm development using a machine-learning framework and

bag-of-features derived from electrocardiogram spectrograms. **Journal of Clinical Medicine**, v. 11, 1 jan. 2022. Citado nas pp. 99, 101, 108, 175, 177.

LIN, Y.; ZHANG, H.; WU, W.; GAO, X.; CHAO, F.; LIN, J. Wavelet transform and deep learning-based obstructive sleep apnea detection from single-lead ECG signals. **Physical and Engineering Sciences in Medicine**, Springer, v. 47, n. 1, p. 119–133, 2024. Citado na p. 110.

LIU, H.; CUI, S.; ZHAO, X.; CONG, F. Detection of obstructive sleep apnea from single-channel ECG signals using a CNN-transformer architecture. **Biomedical Signal Processing and Control**, Elsevier, v. 82, p. 104581, 2023. Citado na p. 110.

LIU, M.-H.; CHIEN, S.-Y.; WU, Y.-L.; SUN, T.-H.; HUANG, C.-S.; HSU, K.-C.; HANG, L.-W. EfficientNet-based machine learning architecture for sleep apnea identification in clinical single-lead ECG signal data sets. **BioMedical Engineering OnLine**, Springer, v. 23, n. 1, p. 57, 2024. Citado na p. 111.

LU, K.-C.; YANG, D.-L. Image processing and image mining using decision trees. **Journal of Information Science & Engineering**, v. 25, n. 4, 2009. Citado na p. 66.

LUNDBERG, S. **SHAP - Welcome to the SHAP documentation**. [S.l.: s.n.]. <https://shap.readthedocs.io/en/latest/>. Software aberto. Versão 0.45.1. Acessado em 2024-02-02. Citado nas pp. 140, 142.

LUNDBERG, S. M.; LEE, S.-I. A unified approach to interpreting model predictions. **Advances in neural information processing systems**, v. 30, 2017. Citado nas pp. 85–88.

LUNDBERG, S. M.; NAIR, B.; VAVILALA, M. S.; HORIBE, M.; EISSES, M. J.; ADAMS, T.; LISTON, D. E.; LOW, D. K.-W.; NEWMAN, S.-F.; KIM, J. et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. **Nature Biomedical Engineering**, v. 2, n. 10, p. 749, 2018. Citado na p. 88.

MAALOUF, M. Logistic regression in data analysis: an overview. **International Journal of Data Analysis Techniques and Strategies**, v. 3, n. 3, p. 281–299, 2011. Citado na p. 55.

MAATEN, L. van der. Learning a Parametric Embedding by Preserving Local Structure. In: PROCEEDINGS of the Twelfth International Conference on Artificial Intelligence and

Statistics. Clearwater Beach, Estados Unidos: [s.n.], 2009. v. 5, p. 384–391. Citado na p. 78.

MCINNES, L. **UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction**. [S.l.: s.n.]. <https://umap-learn.readthedocs.io/en/latest/>. Software aberto. Versão 0.5. Acessado em 2024-02-02. Citado na p. 128.

MCINNES, L.; HEALY, J.; MELVILLE, J. Umap: Uniform manifold approximation and projection for dimension reduction. **arXiv**, 2018. Citado nas pp. 79–81.

MIAO, K.-c.; HAN, T.-t.; YAO, Y.-q.; LU, H.; CHEN, P.; WANG, B.; ZHANG, J. Application of LSTM for short term fog forecasting based on meteorological elements. **Neurocomputing**, v. 408, p. 285–291, 2020. Citado na p. 94.

MIENYE, I. D.; SUN, Y.; WANG, Z. Prediction performance of improved decision tree-based algorithms: a review. In: THE 2nd International Conference on Sustainable Materials Processing and Manufacturing. Sun City, África do Sul: [s.n.], 2019. v. 35, p. 698–703. Citado na p. 65.

MIRÓ-JULIÀ, M.; FIOL-ROIG, G.; ISERN-DEYÀ, A. P. Decision trees in stock market analysis: construction and validation. In: 23RD International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems. Córdoba, Espanha: [s.n.], 2010. P. 185–194. Citado na p. 65.

MOHANDÉS, M.; DERICHE, M.; ALIYU, S. O. Classifiers combination techniques: A comprehensive review. **IEEE Access**, v. 6, p. 19626–19639, 2018. Citado nas pp. 72, 73.

MOKBAL, F. M. M.; DAN, W.; IMRAN, A.; JIUCHUAN, L.; AKHTAR, F.; XIAOXI, W. MLPXSS: An Integrated XSS-Based Attack Detection Scheme in Web Applications Using Multilayer Perceptron Technique. **IEEE Access**, v. 7, p. 100567–100580, 2019. Citado na p. 63.

MOKHTARI, K. E.; HIGDON, B. P.; BAŞAR, A. Interpreting financial time series with SHAP values. In: PROCEEDINGS of the 29th annual international conference on computer science and software engineering. Toronto, Canadá: [s.n.], 2019. P. 166–172. Citado na p. 88.

MOSTAFA, S. S.; MENDONÇA, F.; RAVELO-GARCÍA, A. G.; MORGADO-DIAS, F. A systematic review of detecting sleep apnea using deep learning. **Sensors (Switzerland)**, v. 19, 22 nov. 2019. Citado na p. 30.

MUHAMMAD, I.; YAN, Z. Supervised machine learning approaches: a survey. **ICTACT Journal on Soft Computing**, v. 5, n. 3, 2015. Citado nas pp. 46, 48, 49.

MUKHERJEE, D.; DHAR, K.; SCHWENKER, F.; SARKAR, R. Ensemble of deep learning models for sleep apnea detection: An experimental study. **Sensors**, v. 21, 16 ago. 2021. Citado nas pp. 99, 101, 102, 107, 177.

MURAKAMI, Y.; MIZUGUCHI, K. Applying the Naive Bayes classifier with kernel density estimation to the prediction of protein–protein interaction sites. **Bioinformatics**, v. 26, n. 15, p. 1841–1848, jun. 2010. Citado na p. 54.

MUSHTAQ, Z.; RAMZAN, M. F.; ALI, S.; BASEER, S.; SAMAD, A.; HUSNAIN, M. Voting classification-based diabetes mellitus prediction using hypertuned machine-learning techniques. **Mobile Information Systems**, v. 2022, p. 1–16, 2022. Citado na p. 73.

NAJI, M. A.; EL FILALI, S.; BOUHLAL, M.; BENLAHMAR, E. H.; ABDELOUHAHID, R. A.; DEBAUCHE, O. Breast cancer prediction and diagnosis through a new approach based on majority voting ensemble classifier. **Procedia Computer Science**, v. 191, p. 481–486, 2021. Citado na p. 73.

NASTESKI, V. An overview of the supervised machine learning methods. **Horizons**, v. 4, n. 51-62, p. 56, 2017. Citado na p. 54.

NATEKIN, A.; KNOLL, A. Gradient boosting machines, a tutorial. **Frontiers in neurorobotics**, v. 7, p. 21, 2013. Citado na p. 68.

OTA, R.; YAMASHITA, F. Application of machine learning techniques to the analysis and prediction of drug pharmacokinetics. **Journal of Controlled Release**, v. 352, p. 961–969, 2022. Citado na p. 71.

PADOVANO, D.; MARTINEZ-RODRIGO, A.; PASTOR, J. M.; RIETA, J. J.; ALCARAZ, R. An experimental review on obstructive sleep apnea detection based on heart rate variability and machine learning techniques. In: 2020 8th E-Health and Bioengineering Conference. Iasi, Romênia: [s.n.], out. 2020. Citado nas pp. 100, 101, 106, 177, 179.

PALEYES, A.; URMA, R.-G.; LAWRENCE, N. D. Challenges in Deploying Machine Learning: A Survey of Case Studies. **Association for Computing Machinery Computing Survey**, v. 55, n. 6, dez. 2022. Citado na p. 49.

PARMAR, A.; KATARIYA, R.; PATEL, V. A review on random forest: An ensemble classifier. In: INTERNATIONAL conference on intelligent data communication technologies and internet of things (ICICI) 2018. Coimbatore, Índia: [s.n.], 2019. P. 758–763. Citado na p. 66.

PATTANAYAK, S.; LOHA, C.; HAUCHHUM, L.; SAILO, L. Application of MLP-ANN models for estimating the higher heating value of bamboo biomass. **Biomass Conversion and Biorefinery**, v. 11, p. 2499–2508, 2021. Citado na p. 63.

PAUL, A.; PRASAD, A.; KUMAR, A. Review on artificial neural network and its application in the field of engineering. **Journal of Mechanical Engineering**, v. 1, p. 53–61, 2022. Citado na p. 61.

PAULA, M. C.; CUNHA, L. T.; SILVA, F. N. Síndrome da apneia do sono e seus impactos na saúde: uma revisão integrativa. **Cadernos Camilliani**, v. 17, n. 2, p. 1997–2010, 2020. Citado nas pp. 34–36.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Citado nas pp. 72, 127, 128, 130, 132.

PEDROSA, R. P.; DRAGER, L. F.; GONZAGA, C. C.; SOUSA, M. G.; PAULA, L. K. D.; AMARO, A. C.; AMODEO, C.; BORTOLOTTI, L. A.; KRIEGER, E. M.; BRADLEY, T. D.; LORENZI-FILHO, G. Obstructive sleep apnea: the most common secondary cause of hypertension associated with resistant hypertension. **Hypertension**, v. 58, p. 811–817, 5 nov. 2011. Citado nas pp. 39, 40.

PENZEL, T.; MOODY, G. B.; MARK, R. G.; GOLDBERGER, A. L.; PETER, J. H. The apnea-ECG database. In: COMPUTERS in Cardiology 2000. Vol. 27. Cambridge, Estados Unidos: [s.n.], 2000. P. 255–258. Citado nas pp. 115–117, 119.

PINHO, A.; POMBO, N.; SILVA, B. M.; BOUSSON, K.; GARCIA, N. Towards an accurate sleep apnea detection based on ECG signal: The quintessential of a wise

feature selection. **Applied Soft Computing Journal**, v. 83, out. 2019. Citado nas pp. 100, 101, 106, 177, 179.

PORR, B.; HOWELL, L.; IOANNIS, S.; YOAV, N. **ECG Detectors**. [S.l.: s.n.]. <https://zenodo.org/records/10573363>. Software aberto. Versão 1.3.5. Acessado em 2024-07-27. Citado na p. 122.

PRADO, B. N.; FERNANDES, E. G.; CRISTINA, T.; MOREIRA, A.; GAVRANICH, J. Apneia obstrutiva do sono: diagnóstico e tratamento. **Revista de odontologia da universidade cidade de São Paulo**, v. 22, p. 233–242, 3 2010. Citado nas pp. 34, 39, 41.

PRIYANKA; KUMAR, D. Decision tree classifier: a detailed survey. **International Journal of Information and Decision Sciences**, v. 12, n. 3, p. 246–269, 2020. Citado na p. 65.

PROBST, P.; BOULESTEIX, A.-L.; BISCHL, B. Tunability: Importance of hyperparameters of machine learning algorithms. **The Journal of Machine Learning Research**, v. 20, n. 1, p. 1934–1965, 2019. Citado na p. 82.

QIU, J.; WU, Q.; DING, G.; XU, Y.; FENG, S. A survey of machine learning for big data processing. **EURASIP Journal on Advances in Signal Processing**, v. 2016, p. 1–16, 2016. Citado nas pp. 48, 49.

RAMACHANDRAN, A.; KARUPPIAH, A. A survey on recent advances in machine learning based sleep apnea detection systems. **Healthcare (Switzerland)**, v. 9, 7 jul. 2021. Citado na p. 44.

RAMANESWARAN, S.; SRINIVASAN, K.; VINCENT, P. D. R.; CHANG, C.-Y. Hybrid inception v3 XGBoost model for acute lymphoblastic leukemia classification. **Computational and Mathematical Methods in Medicine**, v. 2021, p. 1–10, 2021. Citado na p. 71.

REDDY, E. M. K.; GURRALA, A.; HASITHA, V. B.; KUMAR, K. V. R. Introduction to Naive Bayes and a review on its subtypes with applications. **Bayesian reasoning and gaussian processes for machine learning applications**, p. 1–14, 2022. Citado na p. 54.

RICHMAN, J. S.; MOORMAN, J. R. Physiological time-series analysis using approximate entropy and sample entropy. **American journal of physiology-heart and circulatory physiology**, 2000. Citado na p. 126.

ROCHA, C. S. C. da. **Técnicas de reconhecimento de padrões para a classificação de sinais fisiológicos**. 2013. Tese (Doutorado) – Universidade Federal de Minas Gerais, Belo Horizonte, Minas Gerais, Brasil. Citado nas pp. 34–36.

ROPKE, L. M.; SOUZA, A. G.; MAGALHÃES BERTOZ, A. P. de; ADRIAZOLA, M. M.; ORTOLAN, E. V. P.; MARTINS, R. H.; LOPES, W. C.; RODRIGUES, C. D. B.; BIGLIAZZI, R.; WEBER, S. A. T. Efeito da atividade física na qualidade do sono e qualidade de vida: revisão sistematizada. **Archives of Health Investigation**, v. 6, n. 12, 2017. Citado na p. 36.

ROUSSEEUW, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. **Journal of Computational and Applied Mathematics**, v. 20, p. 53–65, 1987. Citado na p. 128.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach, Global Edition, 4th Foundations, vol. 19**. 4. ed. Harlow, Reino Unido: Pearson Education, 2021. Citado nas pp. 30, 44, 46, 50, 53, 54, 56–58, 60, 61, 63, 72, 73, 76, 88, 90.

SADR, N.; CHAZAL, P. D. A comparison of three ECG-derived respiration methods for sleep apnoea detection. **Biomedical Physics and Engineering Express**, v. 5, 2 jan. 2019. Citado nas pp. 100, 101, 177, 178.

SADR, N.; CHAZAL, P. D.; SCHAİK, A. V.; BREEN, P. Sleep apnoea episodes recognition by a committee of ELM classifiers from ECG signal. In: PROCEEDINGS of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Milão, Itália: [s.n.], nov. 2015. 2015-November, p. 7675–7678. Citado nas pp. 100, 101, 106, 177.

SALIH HASAN, B. M.; ABDULAZEEZ, A. M. A review of principal component analysis algorithm for dimensionality reduction. **Journal of Soft Computing and Data Mining**, v. 2, n. 1, p. 20–30, 2021. Citado nas pp. 74, 75.

AL-SALIHY, N. K.; IBRIKCI, T. Classifying breast cancer by using decision tree algorithms. In: PROCEEDINGS of the 6th International Conference on Software and Computer Applications. Bancoque, Tailândia: [s.n.], 2017. P. 144–148. Citado na p. 66.

SCHAPIRE, R. E. Explaining adaboost. In: EMPIRICAL Inference: Festschrift in Honor of Vladimir N. Vapnik. Berlin, Alemanha: Springer Berlin Heidelberg, 2013. P. 37–52. Citado na p. 68.

SETHIA, A.; RAUT, P. Application of LSTM, GRU and ICA for stock price prediction. In: INFORMATION and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2018, Volume 2. Pandangue, Indonésia: [s.n.], 2019. P. 479–487. Citado na p. 94.

SHAHRIARI, B.; SWERSKY, K.; WANG, Z.; ADAMS, R. P.; DE FREITAS, N. Taking the human out of the loop: A review of Bayesian optimization. **Proceedings of the IEEE**, v. 104, n. 1, p. 148–175, 2015. Citado na p. 84.

SHAMSHIRBAND, S.; FATHI, M.; DEHZANGI, A.; CHRONOPOULOS, A. T.; ALINEJAD-ROKNY, H. A review on deep learning approaches in healthcare systems: Taxonomies, challenges, and open issues. **Journal of Biomedical Informatics**, v. 113, p. 103627, 2021. Citado na p. 89.

SHARAN, R. V.; BERKOVSKY, S.; XIONG, H.; COIERA, E. End-to-End Sleep Apnea Detection Using Single-Lead ECG Signal and 1-D Residual Neural Networks. **Journal of Medical and Biological Engineering**, v. 41, p. 758–766, 5 out. 2021. Citado nas pp. 99, 101, 107, 177.

SHETA, A.; TURABIEH, H.; THAHER, T.; TOO, J.; MAFARJA, M.; HOSSAIN, M. S.; SURANI, S. R. Diagnosis of obstructive sleep apnea from ECG signals using machine learning and deep learning classifiers. **Applied Sciences (Switzerland)**, v. 11, 14 jul. 2021. Citado nas pp. 30, 99, 101, 107.

SHNEERSON, J. M. **Sleep medicine: a guide to sleep and its disorders**. 2. ed. Oxford, Estados Unidos: John Wiley & Sons, 2009. Citado nas pp. 29, 34–40, 126.

SHRADHA VERMA, A. C.; SINGH, A. P. Application of convolutional neural networks for evaluation of disease severity in tomato plant. **Journal of Discrete Mathematical Sciences and Cryptography**, v. 23, n. 1, p. 273–282, 2020. Citado na p. 92.

SMAGULOVA, K.; JAMES, A. P. A survey on LSTM memristive neural network architectures and applications. **The European Physical Journal Special Topics**, v. 228, n. 10, p. 2313–2324, 2019. Citado na p. 94.

SOMVANSHI, M.; CHAVAN, P.; TAMBADE, S.; SHINDE, S. V. A review of machine learning techniques using decision tree and support vector machine. In: 2016 International Conference on Computing Communication Control and automation. Pune, Índia: [s.n.], 2016. P. 1–7. Citado na p. 64.

SRINIVAS, P.; KATARYA, R. hyOPTXg: OPTUNA hyper-parameter optimization framework for predicting cardiovascular disease using XGBoost. **Biomedical Signal Processing and Control**, v. 73, p. 103456, 2022. Citado na p. 84.

SRINIVASULU, A.; MOHAN, S.; HARIKA, T.; SRUJANA, P.; REVATHI, Y. Apnea event detection using machine learning technique for the clinical diagnosis of sleep apnea syndrome. In: 2021 3rd International Conference on Signal Processing and Communication. Coimbatore, Índia: Institute of Electrical e Electronics Engineers Inc., mai. 2021. P. 490–493. Citado nas pp. 99, 101, 107.

SRIVASTAVA, G.; CHAUHAN, A.; KARGETI, N.; PRADHAN, N.; DHAKA, V. S. ApneaNet: A hybrid 1DCNN-LSTM architecture for detection of Obstructive Sleep Apnea using digitized ECG signals. **Biomedical Signal Processing and Control**, Elsevier, v. 84, p. 104754, 2023. Citado na p. 110.

STRZELECKA, A.; KURDYS-KUJAWSKA, A.; ZAWADZKA, D. Application of logistic regression models to assess household financial decisions regarding debt. In: KNOWLEDGE-BASED and Intelligent Information and Engineering Systems: Proceedings of the 24th International Conference KES2020. Online: [s.n.], 2020. v. 176, p. 3418–3427. Citado na p. 56.

SUMARSONO, A.; MASTERS, Z. Application of LSTM Auto Encoder in Hardware Trojan Detection. In: 2023 IEEE 13th Annual Computing and Communication Workshop and Conference. Las Vegas, Estados Unidos: [s.n.], 2023. P. 566–571. Citado na p. 94.

SYARIF, I.; ZALUSKA, E.; PRUGEL-BENNETT, A.; WILLS, G. Application of bagging, boosting and stacking to intrusion detection. In: MACHINE Learning and Data Mining in Pattern Recognition: 8th International Conference. Proceedings 8. Berlim, Alemanha: [s.n.], 2012. P. 593–602. Citado na p. 73.

THARWAT, A. Principal component analysis-a tutorial. **International Journal of Applied Pattern Recognition**, v. 3, n. 3, p. 197–240, 2016. Citado na p. 75.

THOMPSON, S.; REILLY, D.; FERGUS, P.; CHALMERS, C. Detection of Obstructive Sleep Apnoea Using Features Extracted From Segmented Time-Series ECG Signals With a One Dimensional Convolutional Neural Network. **IEEE Access**, IEEE, 2023. Citado na p. 110.

VAN DER MAATEN, L.; HINTON, G. Visualizing data using t-SNE. **Journal of machine learning research**, v. 9, n. 11, 2008. Citado nas pp. 76, 78, 81.

VAN RIJN, J. N.; HUTTER, F. Hyperparameter importance across datasets. In: PROCEEDINGS of the 24th Association for Computing Machinery Special Interest Group on Knowledge Discovery in Data International Conference on Knowledge Discovery & Data Mining. Nova Iorque, Estados Unidos: [s.n.], 2018. P. 2367–2376. Citado na p. 82.

VEASEY, S. C.; ROSEN, I. M. Obstructive Sleep Apnea in Adults. **New England Journal of Medicine**, v. 380, p. 1442–1449, 15 abr. 2019. Citado na p. 29.

VIAENE, S.; DERRIG, R. A.; DEDENE, G. A case study of applying boosting Naive Bayes to claim fraud diagnosis. **IEEE Transactions on Knowledge and Data Engineering**, v. 16, n. 5, p. 612–620, 2004. Citado na p. 54.

WAGLE, N.; ACHARYA, T. D.; LEE, D. H. Comprehensive review on application of machine learning algorithms for water quality parameter estimation using remote sensing data. **Sensors & Materials**, v. 32, 2020. Citado na p. 71.

WANG, Z.; PAN, X.; MEI, Z.; XU, Z.; LV, Y.; ZHANG, Y.; GUAN, C. ECGAN-Assisted ResT-Net Based on Fuzziness for OSA Detection. **IEEE Transactions on Biomedical Engineering**, IEEE, 2024. Citado na p. 111.

WASKOM, M. **Seaborn**. [S.l.: s.n.]. <https://seaborn.pydata.org/index.html>. Software aberto. Versão 0.13. Acessado em 2024-07-27. Citado na p. 142.

WEERTS, H. J.; MUELLER, A. C.; VANSCHOREN, J. Importance of tuning hyperparameters of machine learning algorithms. **arXiv**, 2020. Citado na p. 82.

WICKRAMASINGHE, I.; KALUTARAGE, H. Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. **Soft Computing**, v. 25, n. 3, p. 2277–2293, 2021. Citado nas pp. 53, 54.

WONG, W.-T.; HSU, S.-H. Application of SVM and ANN for image retrieval. **European Journal of Operational Research**, v. 173, n. 3, p. 938–950, 2006. Citado na p. 59.

WONGSIRICHOT, T.; HANSKUNATAI, A. A classification of sleep disorders with optimal features using machine learning techniques. **Journal of Health Research**, v. 31, 3 2017. Citado nas pp. 100, 101, 106.

WU, Y.-c.; FENG, J.-w. Development and application of artificial neural network. **Wireless Personal Communications**, v. 102, p. 1645–1656, 2018. Citado nas pp. 60, 61, 63.

XIE, C.; MCCULLUM, L.; JOHNSON, A.; POLLARD, T.; GOW, B.; MOODY, B. **Waveform Database Software Package (WFDB) for Python**. [S.l.: s.n.]. <https://physionet.org/content/wfdb-python/4.1.0/>. Software aberto. Versão 4.1.0. Acessado em 2024-07-27. Citado na p. 122.

YANG, C.; CHEN, M.; YUAN, Q. The application of XGBoost and SHAP to examining the factors in freight truck-related crashes: An exploratory analysis. **Accident Analysis & Prevention**, v. 158, p. 106153, 2021. Citado na p. 88.

YANG, Q.; ZOU, L.; WEI, K.; LIU, G. Obstructive sleep apnea detection from single-lead electrocardiogram signals using one-dimensional squeeze-and-excitation residual group network. **Computers in Biology and Medicine**, v. 140, jan. 2022. Citado nas pp. 30, 99, 101, 107, 115, 175, 177.

YANG, S.; YU, X.; ZHOU, Y. LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example. In: 2020 International Workshop on Electronic Communication and Artificial Intelligence. [S.l.: s.n.], 2020. P. 98–101. Citado na p. 94.

YEO, M.; BYUN, H.; LEE, J.; BYUN, J.; RHEE, H.-Y.; SHIN, W.; YOON, H. Robust method for screening sleep apnea with single-lead ecg using deep residual network: evaluation with open database and patch-type wearable device data. **IEEE Journal of Biomedical and Health Informatics**, IEEE, v. 26, n. 11, p. 5428–5438, 2022. Citado na p. 110.

YU, Y.; SI, X.; HU, C.; ZHANG, J. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. **Neural Computation**, v. 31, n. 7, p. 1235–1270, jul. 2019. Citado nas pp. 93, 94.

ZAREAPOOR, M.; SHAMSOLMOALI, P. et al. Application of credit card fraud detection: Based on bagging ensemble classifier. **Procedia computer science**, v. 48, n. 2015, p. 679–685, 2015. Citado na p. 73.

ZHU, Z.; ZHOU, M.; HU, F.; WANG, S.; MA, J.; GAO, B.; BIAN, K.; LAI, W. A day-ahead industrial load forecasting model using load change rate features and combining FA-ELM and the AdaBoost algorithm. **Energy Reports**, v. 9, p. 971–981, 2023. Citado na p. 68.

## APÊNDICE 1 – RESULTADOS DOS MODELOS APRENDIZADO DE MÁQUINAS E APRENDIZADO PROFUNDO

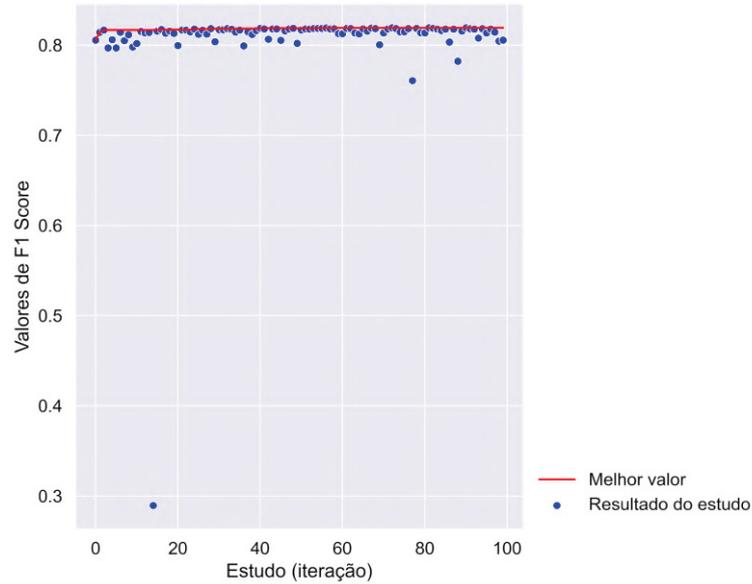
Neste capítulo são apresentados e brevemente discutido os resultados dos modelos de ML que não foram apresentados nas seção 6.2 ou no capítulo 7, portando é recomendo a leitura prévia desta seção e capítulo antes de analisar os demais resultados aqui expostos. Em cada seção deste capítulo é apresentado os resultados do histórico de otimização, o impacto dos hiperparâmetros no desempenho e no tempo de otimização, as características que mais contribuíram para explicar o resultado obtido e a matriz de confusão para cada um dos modelos. É seguida a mesma ordem da seção 6.2, com a exclusão do modelo de SVM, cujo todos os resultados foram apresentados na subseção 6.2.3.1. Por fim, é apresentado o resultado dos testes de hipóteses dos experimentos.

### DEMAIS RESULTADOS DO *ADAPTATIVE BOOSTING*, ADABOOST

As FIGURAS 60, 61, 62 e 63 mostram o histórico de evolução do AdaBoost para cada um dos experimentos. Analisando as FIGURAS, as configurações testadas no experimento I apresentaram resultados na base de validação mais próximos, com um *outlier* que teve um desempenho inferior. Este comportamento também é visto no experimento III, mas que além do *outlier*, as configurações tiveram uma maior variabilidade de desempenho do que no experimento I. Enquanto que nos experimento II e IV, não houveram *outliers*, e os resultados ficaram mais próximos. A FIGURA 64 exhibe dois gráficos, em que o gráfico de cima (indicado por "a") mostra a importância relativa de cada hiperparâmetro no desempenho do modelo e o gráfico de baixo, indicado por "b" o impacto relativo do hiperparâmetro no tempo de otimização. Enquanto que a taxa de aprendizado teve mais impacto no desempenho do modelo em todos os experimentos, o maior impacto no tempo foi causado pelo hiperparâmetro número de estimadores. Isso pode ter acontecido, pois a taxa de aprendizado permite que os ajustes nas funções dos estimadores sejam feitas de uma forma mais suave, evitando causar um *overfitting* ou *underfitting*, enquanto que o número de estimadores, significa o total de classificadores que serão treinados durante aquela iteração da otimização, sendo esta a atividade que mais consome tempo na modelagem por ml.

Ao analisar a explicabilidade de cada característica no resultado do modelo AdaBoost, as três primeiras componentes do PCA contribuíram com maior média de importância, conforme observado na FIGURA 65, com destaque para segunda componente. Na FIGURA 66 a componente de número dois teve um papel mais significativo na explicabilidade do resultado do experimento III enquanto que na FIGURA 67 mostra que

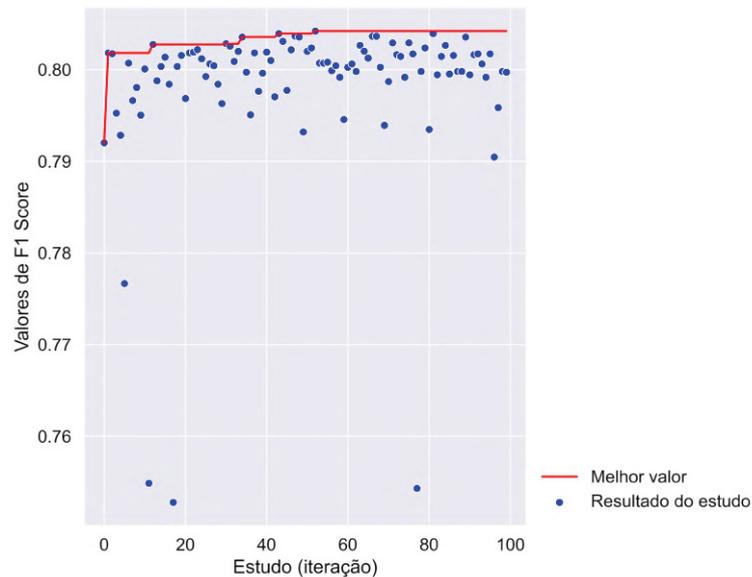
FIGURA 60 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO ADABOOST NO EXPERIMENTO I



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de AdaBoost no experimento I.

FIGURA 61 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO ADABOOST NO EXPERIMENTO II



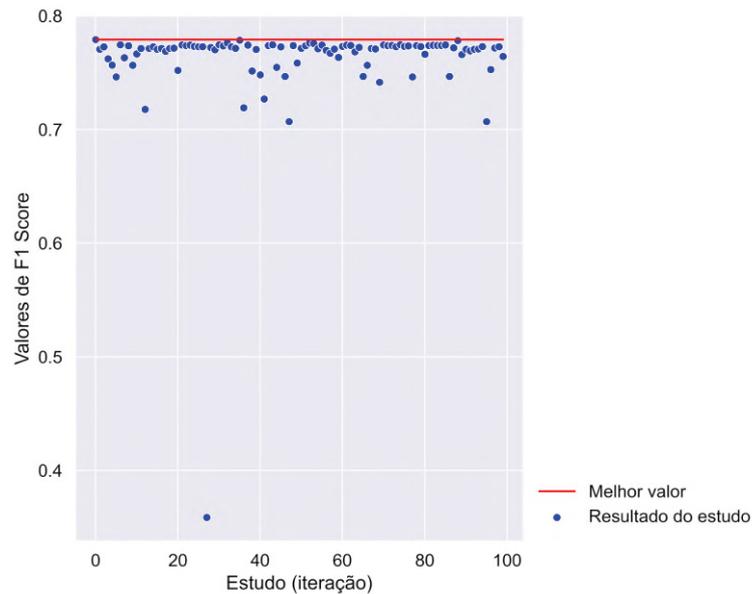
FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de AdaBoost no experimento II.

o maior impacto na explicabilidade veio da componente de número três no experimento IV. Os resultados referentes ao experimento I são apresentados na subseção 6.2.2.2.

As FIGURAS 68, 69 e 70 mostram o processo decisório, através da análise

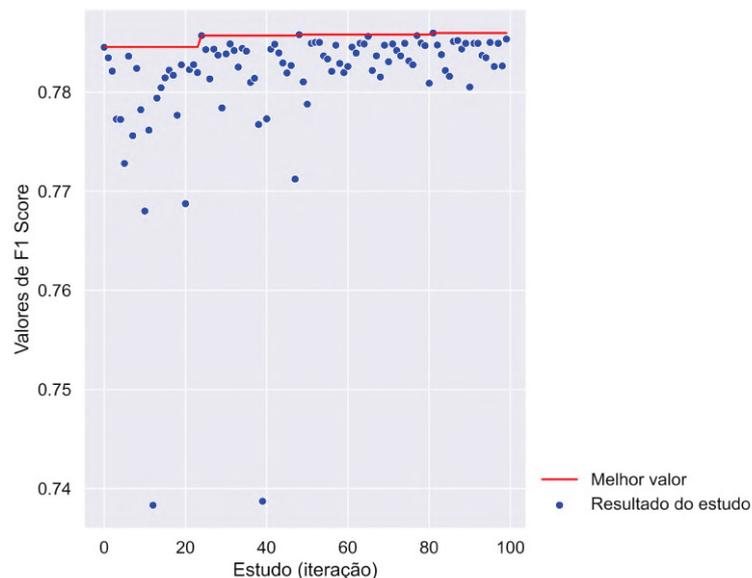
FIGURA 62 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO ADABOOST NO EXPERIMENTO III



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de AdaBoost no experimento III.

FIGURA 63 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO ADABOOST NO EXPERIMENTO IV

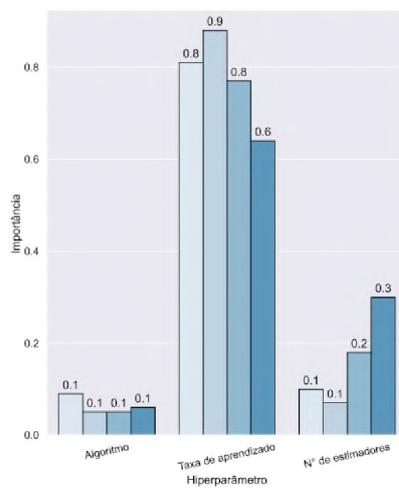


FONTE: Próprio autor

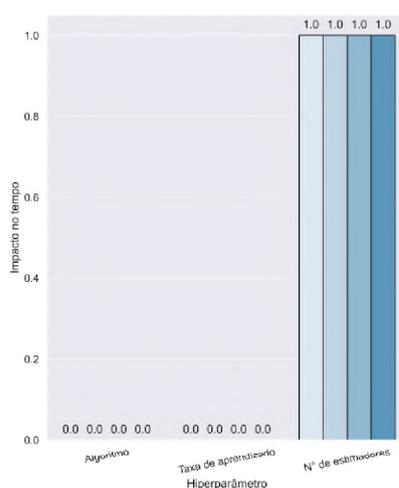
LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de AdaBoost no experimento IV.

de valores SHAP, para as 200 amostras selecionadas nos experimentos II, III e IV (os resultados do experimento I são mostrados na subseção 6.2.2.2). É possível perceber que à medida que a quantidade de dimensões diminui, as linhas de predição tendem a

FIGURA 64 – IMPACTOS DOS HIPERPARÂMETROS NA OTIMIZAÇÃO DO MODELO ADA-BOOST



(a) importância relativa no desempenho



(b) impacto relativo no tempo de otimização

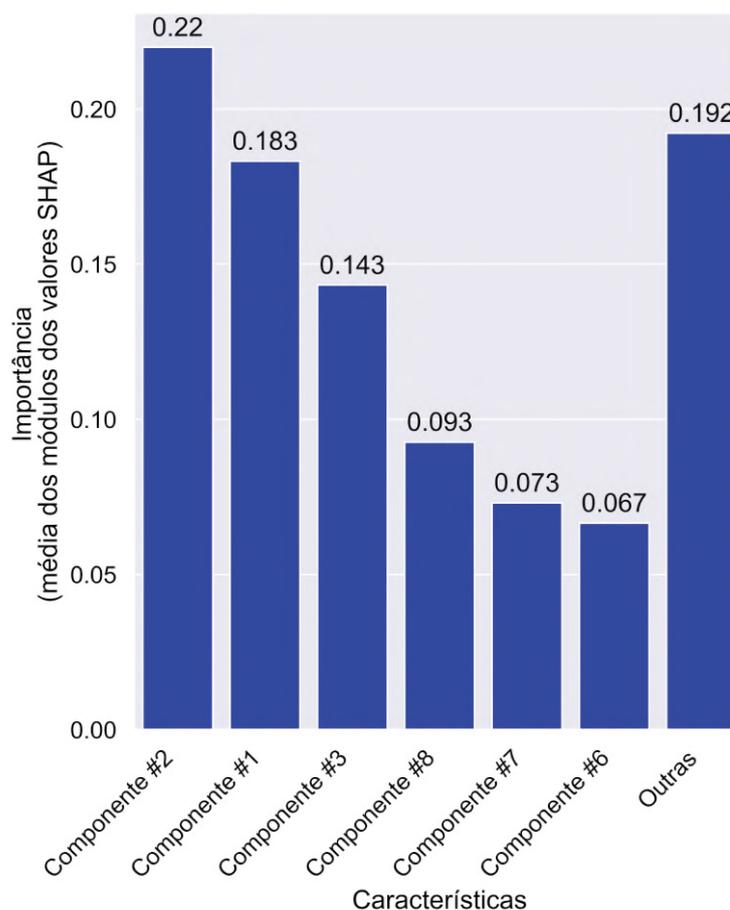
Experimentos  
 sem redução de dimensionalidade  
 PCA  
 t-SNE  
 UMAP

FONTE: Próprio autor  
 LEGENDA: Impactos dos hiperparâmetros no desempenho e no tempo de otimização para modelo AdaBoost.

ficarem mais homogêneas. No experimento II, as componentes de menor impacto na predição do modelo parecem ter uma influência grande em alguns erros do modelo, enquanto que nos experimentos III e IV, este padrão é mais difícil de se observar, devido a quantidade de componentes ser apenas três, no entanto, alguns valores de cada uma das componentes parece desviar para erros grandes.

A FIGURA 71 apresenta a matriz de confusão para o modelo AdaBoost. Na FIGURA, é possível observar que à medida que o número de dimensões diminui, o modelo erra mais ao predizer eventos de AOS. Enquanto que para eventos de não-AOS,

FIGURA 65 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO ADABOOST NO EXPERIMENTO II



FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo AdaBoost no experimento II.

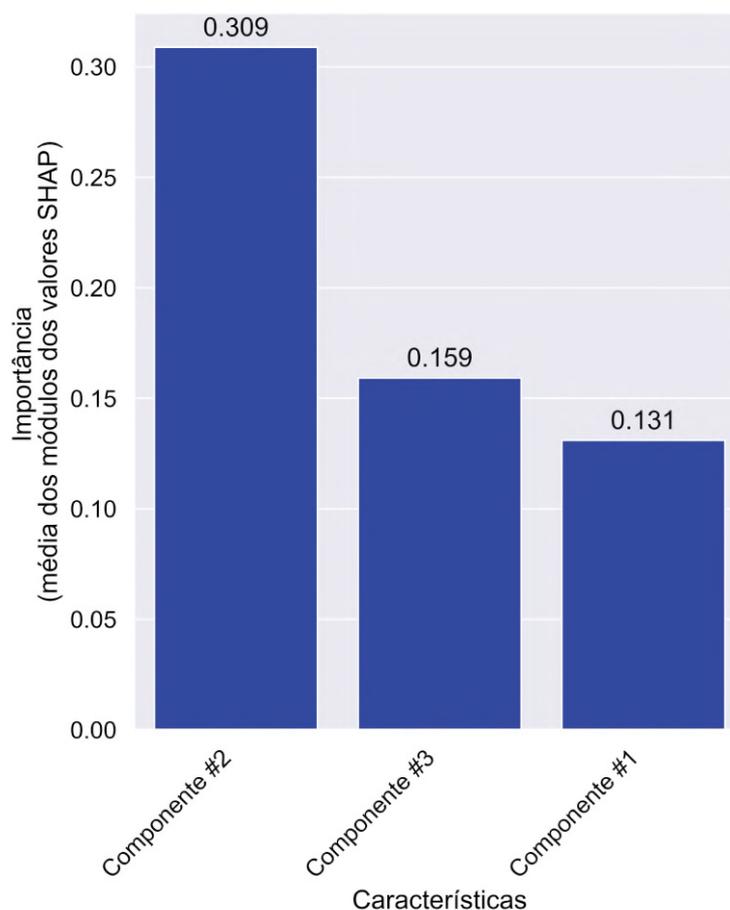
a acuracidade do modelo segue parecida, com a exceção do experimento III.

#### DEMAIS RESULTADOS DAS REDES NEURAIIS DO TIPO MULTICAMADA DE *PERCEPTRONS*, MLP

As FIGURAS 72, 73, 74 e 75 mostram o histórico de otimização do modelo de redes neurais MLP. No geral, as configurações testadas do MLP em todos os experimentos diferiram entre 0,05 e 0,06 de F1-score em todos os experimentos, com os experimentos com t-SNE e UMAP apresentando duas configurações resultados inferiores as demais. De forma geral, as configurações testadas de MLP apresentaram um resultado estável.

Analisando o impacto de cada hiperparâmetro do modelo MLP no desempenho durante o treinamento e no tempo total de otimização na FIGURA 76, de forma geral, é possível observar que, com a exceção do experimento IV, a quantidade de unidades

FIGURA 66 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO ADABOOST NO EXPERIMENTO III

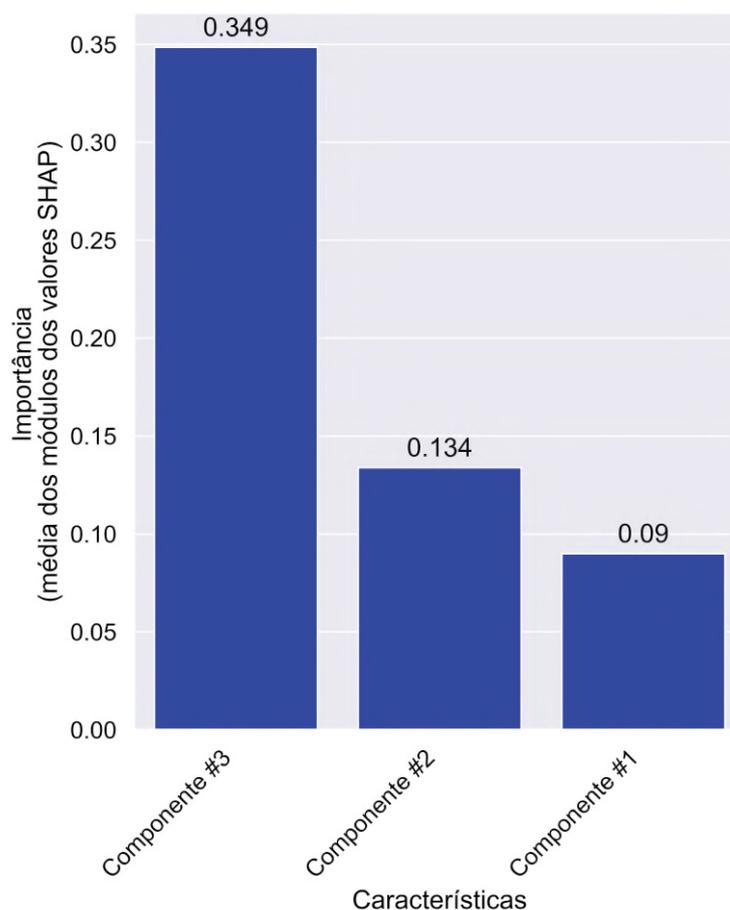


FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo AdaBoost no experimento III.

na camada 1 da rede neural mais impactou o desempenho. Para o experimento IV, a taxa de aprendizado foi o hiperparâmetro que mais impactou o desempenho. O número de camadas, não foi um hiperparâmetro importante, no geral, para nenhum dos experimentos. Quanto ao impacto do tempo, houve uma variância maior no que cada hiperparâmetro impactou no tempo de otimização. Para experimento com aplicação do t-SNE, a taxa de aprendizado foi o que mais impactou o tempo. Para os demais, houve um maior equilíbrio no impacto dos hiperparâmetros do número de camadas, unidades na primeira camada, função de ativação e taxa de aprendizado. As demais camadas do modelo MLP não impactaram muito o tempo e nem o desempenho, isto pode ter acontecido caso a otimização bayesiana do Optuna tenha sido dirigida para um caminho de testar mais configurações com uma única camada, ou com mais variações na primeira camada. Uma análise similar ao realizado com a SVM, descrito na subseção 6.2.3.1, em que foram levantados as configurações de um hiperparâmetro testadas e a média de resultados obtidos, pode ajudar entender melhor o comportamento do MLP

FIGURA 67 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO ADABOOST NO EXPERIMENTO IV



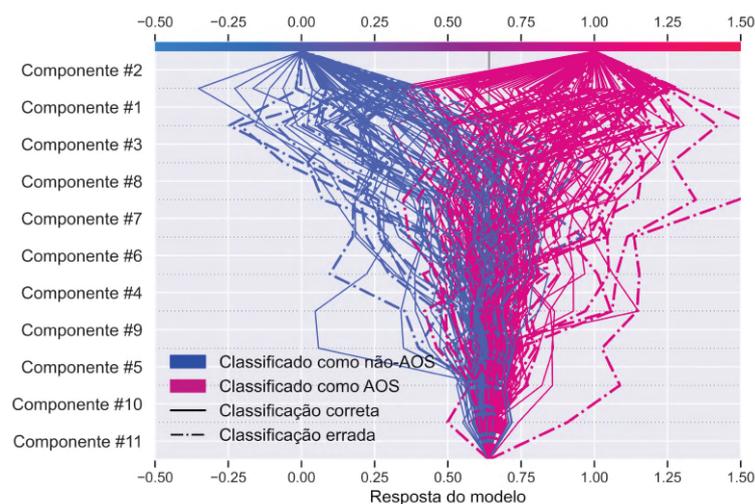
FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo AdaBoost no experimento IV.

neste problema. Esta análise ajuda a isolar uma configuração específica e como esta configuração pode ter impactado no resultado.

Quanto à explicabilidade do modelo MLP, as importâncias média de cada característica ou componente são mostradas nas FIGURAS 77, 78, 79 e 80 para cada um dos experimentos. Na FIGURA 77 é mostrado a importância de cada característica no experimento sem redução de dimensionalidade. Nesta FIGURA, é possível observar que não houve um conjunto de características específico que foi mais determinante na explicação do resultado do modelo, mas sim um certo balanceamento entre todas as 23 características. O que, de certa forma, condiz com o comportamento de uma rede neural, que cria conexões e pesos para cada uma das entradas da rede. Na FIGURA 78, do experimento utilizando PCA, as componentes 1 e 2 foram as que tiveram maior importância, embora ainda exista um certo equilíbrio entre as componentes, similar ao observado no experimento sem redução de dimensionalidade. No experimento do PCA, a componente 7 ficou em terceiro lugar de importância, um achado interessante,

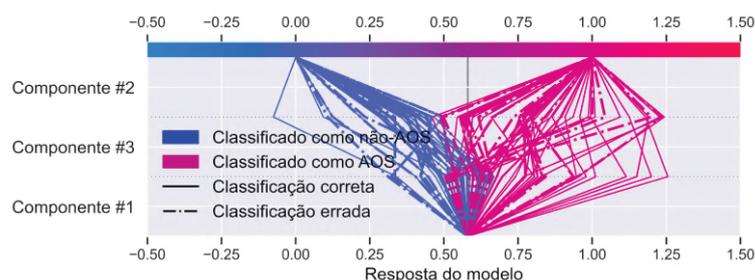
FIGURA 68 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO ADABOOST NO EXPERIMENTO II



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo AdaBoost no experimento II.

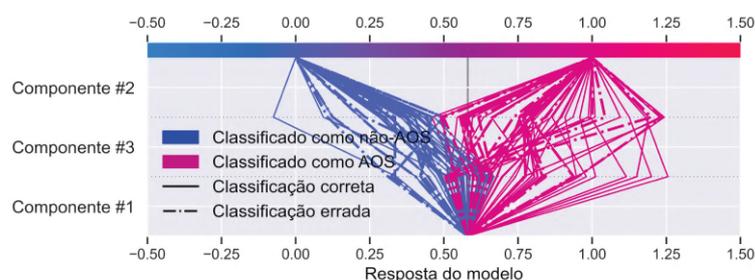
FIGURA 69 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO ADABOOST NO EXPERIMENTO III



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo AdaBoost no experimento III.

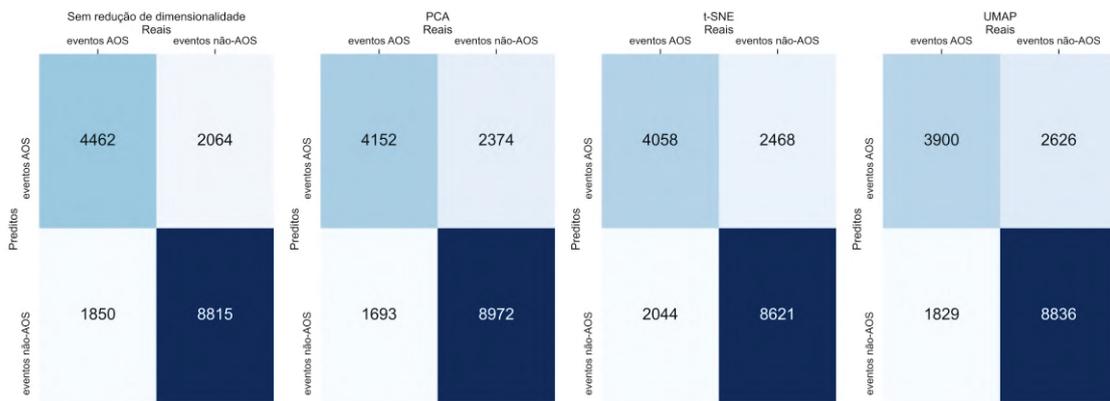
FIGURA 70 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO ADABOOST NO EXPERIMENTO IV



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo AdaBoost no experimento IV.

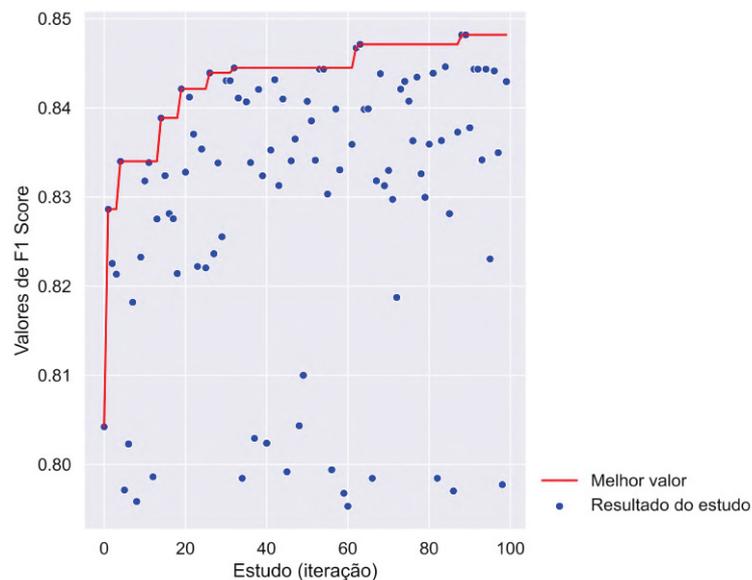
FIGURA 71 – MATRIZ DE CONFUSÃO DO MODELO ADABOOST



FONTE: Próprio autor

LEGENDA: Matriz de confusão para o modelo AdaBoost em cada um dos experimentos.

FIGURA 72 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO MLP NO EXPERIMENTO I



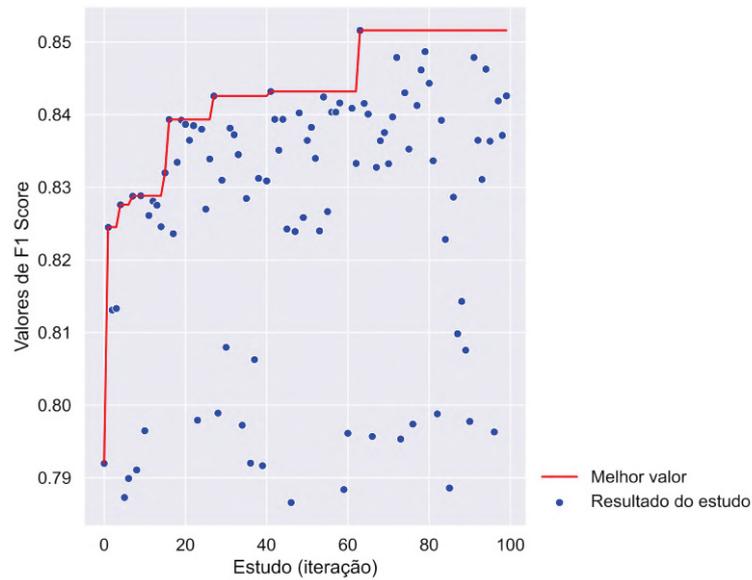
FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de MLP no experimento I.

uma vez que na redução com PCA, está seria a sétima componente em termos de quantidade de variância do conjunto original de dados. A FIGURA 79 mostra que as componentes do experimento com aplicação utilizando t-SNE também tiveram um comportamento de equilíbrio em termos de importância, enquanto que na FIGURA 80, do experimento UMAP, a componente 3 teve um valor de importância maior, enquanto a primeira componente teve o menor valor.

A FIGURA 81 mostra o impacto de cada característica no processo de predição de cada classe para as 200 amostras no experimento I. Na FIGURA é possível observar que algumas características, como pNN50, NN50, pNN20 e CVNNI adicionam um peso alto na resposta do modelo, independente das classes e de acertos ou erros do

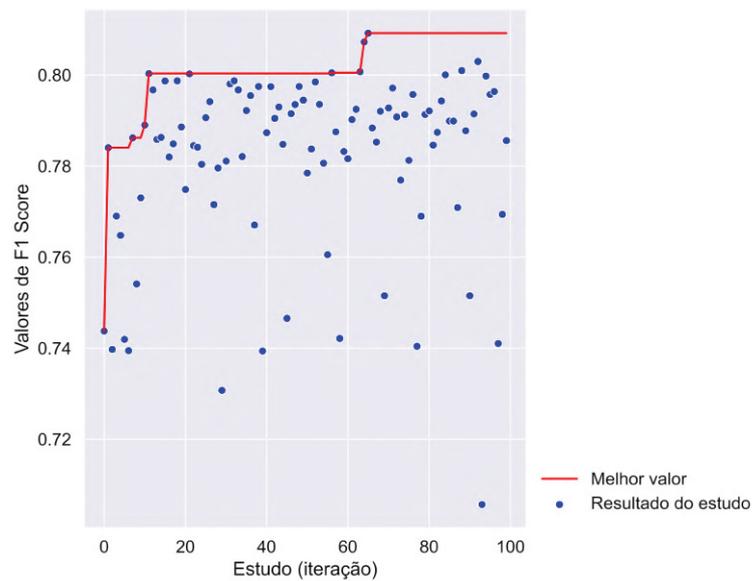
FIGURA 73 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO MLP NO EXPERIMENTO II



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de MLP no experimento II.

FIGURA 74 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO MLP NO EXPERIMENTO III

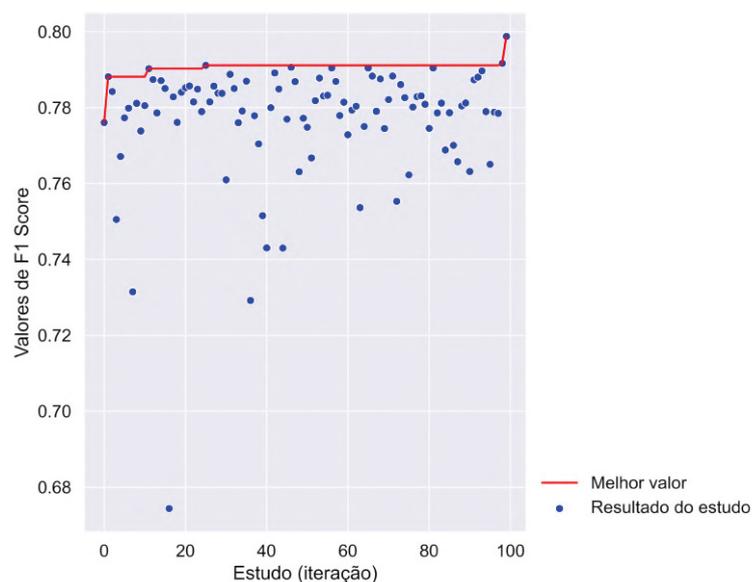


FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de MLP no experimento III.

modelo. Uma análise mais a fundo dos valores relativos estas características podem ajudar entender este impacto. Outro ponto a ser ressaltado na FIGURA é o valor inicial da resposta do modelo, uma espécie de viés, igual a aproximadamente 0,63, indicando que o modelo está levemente inclinado a classificar eventos como AOS. A FIGURA 82 mostra o mesmo processo para o experimento II. Com a utilização do

FIGURA 75 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO MLP NO EXPERIMENTO IV



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de MLP no experimento IV.

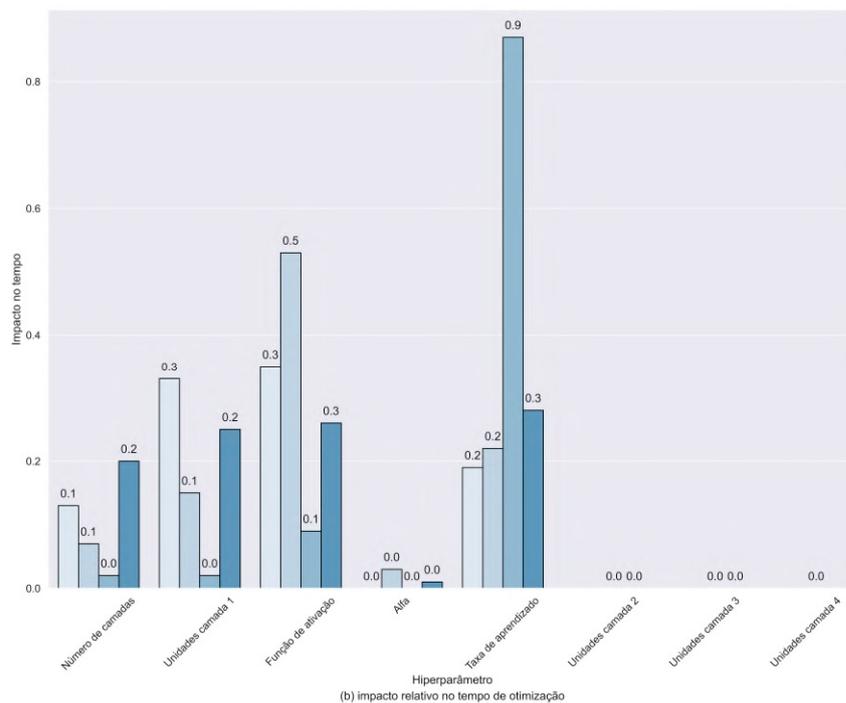
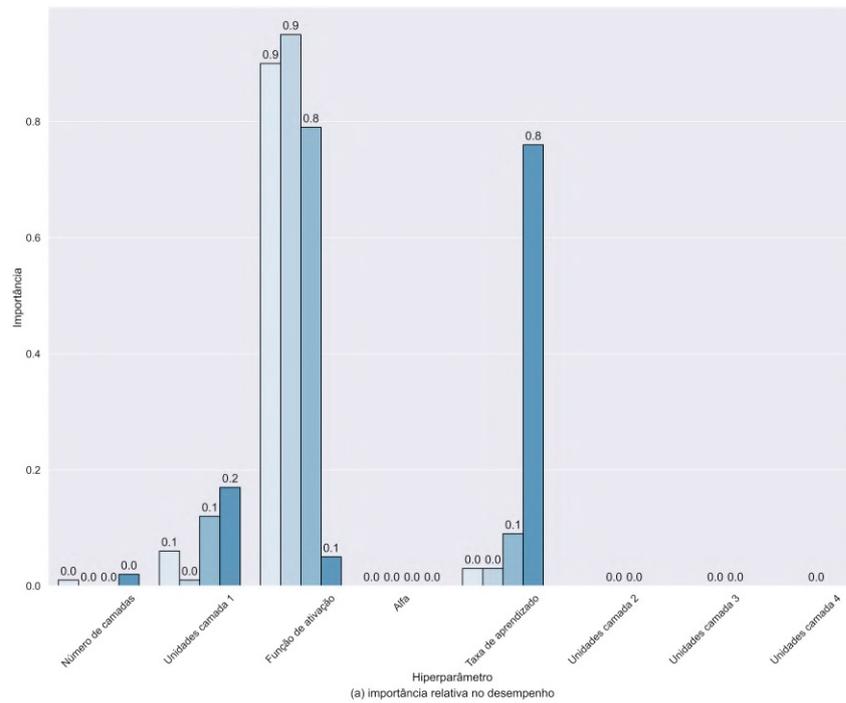
PCA, como mostra na FIGURA, o valor inicial do modelo é menor, em cerca de 0,55. É possível analisar que as componentes que têm menor poder de explicabilidade, 10 e 11, influenciaram fortemente alguns erros de classificação do tipo I e II. À medida que as dimensões diminuíram nos experimentos III e IV, mostrados nas FIGURAS 83 e 84 respectivamente, as linhas de predições ficaram mais homogêneas.

A FIGURA 85 apresenta as matrizes de confusão para os experimentos com MLP. Observando a FIGURA, é possível analisar que houve um aumento de cerca de 500 eventos de AOS classificados corretamente do experimento I para experimento II. O experimento IV teve maior número de acertos de eventos não-AOS, enquanto o experimento III apresentou o maior erro em eventos de AOS.

#### DEMAIS RESULTADOS DO MODELO *BAGGING*

As FIGURAS 86, 87, 88 e 89 mostram o histórico de otimização para o modelo *Bagging*. Na FIGURA 86, do experimento sem redução de dimensionalidade, é possível observar três diferentes faixas de valores de F1-score na base de validação para diferentes configurações do modelo. Para o experimento utilizando PCA, na FIGURA 87, é possível observar duas faixas diferentes, uma mais larga e com resultados melhores e outra menor com resultados piores com uma diferença entre 0,05 e 0,03 de F1-score. Algo similar é observado na FIGURA 88, do experimento com t-SNE, onde existem duas faixas de resultado e na faixa mais ampla, é possível dividi-la. O comportamento menos diverso é observado na FIGURA 89, com as configurações

FIGURA 76 – IMPACTOS DOS HIPERPARÂMETROS NA OTIMIZAÇÃO DO MODELO MLP

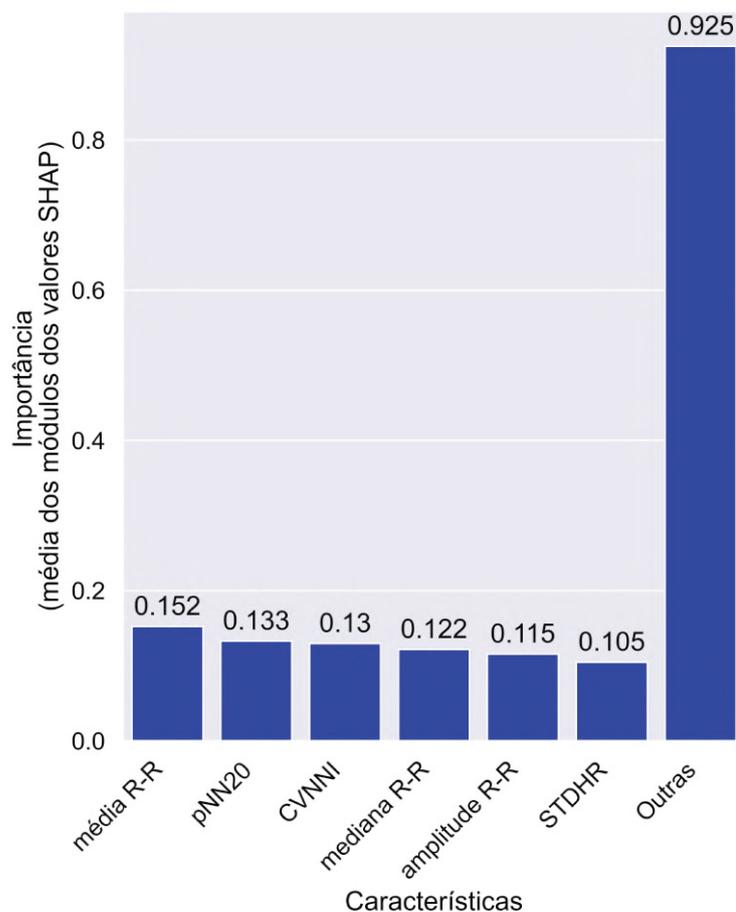


Experimentos  
 sem redução de dimensionalidade  
 PCA  
 t-SNE  
 UMAP

FONTE: Próprio autor  
 LEGENDA: Impactos dos hiperparâmetros no desempenho e no tempo de otimização para modelo MLP.

variando em resultado em no máximo 0,02. Este comportamento do modelo *bagging* pode ser devido ao fato que, mesmo com diferentes configurações, os resultados eram

FIGURA 77 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO MLP NO EXPERIMENTO I



FONTE: Próprio autor

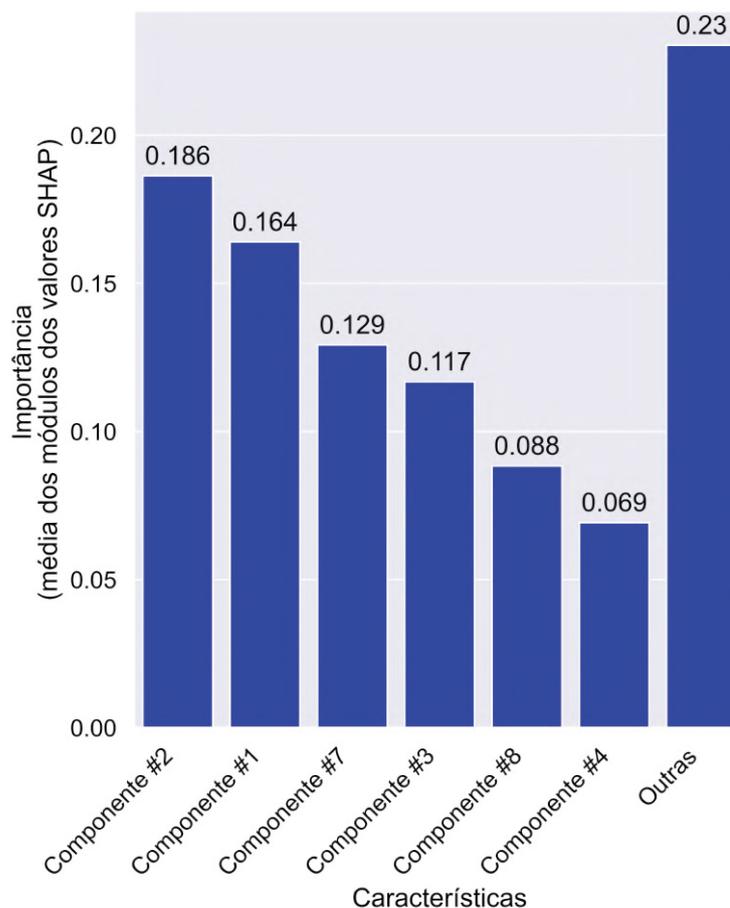
LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo MLP no experimento I.

direcionados por um ou dois hiperparâmetros chaves. Os próximos gráficos podem ajudar a elucidar este ponto.

A FIGURA 90 apresenta a importância relativa de cada hiperparâmetro do resultado durante o treinamento em cada experimento, no gráfico da direita, e o impacto relativo de cada hiperparâmetro no tempo de otimização em cada experimento. Conforme teorizado, o hiperparâmetro do tipo de classificador foi o que mais impactou o resultado dos modelos, em quase que na totalidade, para todos os experimentos. Também foi o hiperparâmetro que mais impactou no tempo de otimização, seguido pelo total de estimadores.

Para o modelo *bagging*, não houve análise de explicabilidade. A FIGURA 91 apresenta as matrizes de confusão do modelo em cada um dos experimentos. É possível observar que o modelo errou mais à medida que o número de dimensões diminuiu. Isto pode ter acontecido devido ao fato de um modelo de *ensemble* como o

FIGURA 78 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO MLP NO EXPERIMENTO II



FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo MLP no experimento II.

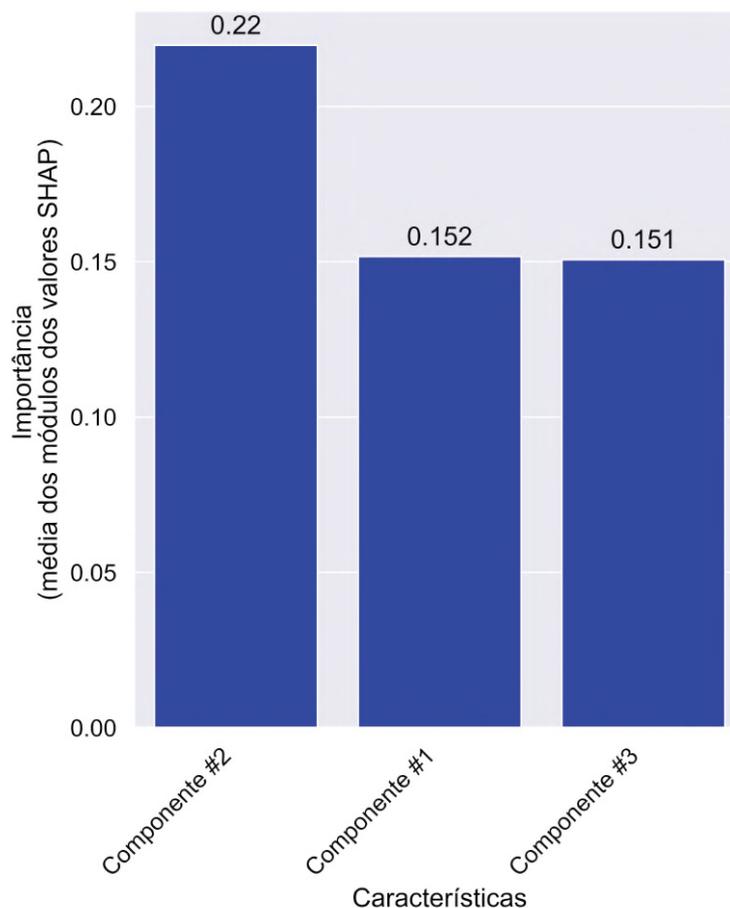
*bagging* pode ser capaz de criar diferentes classificadores para cada tipo de padrão na base de dados, logo, quanto mais informação disponível, mais pode ajudar na separação dos eventos. Uma análise mais profunda da quantidade de classificadores e tipo pode ajudar na análise deste comportamento.

#### DEMAIS RESULTADOS DAS ÁRVORES DE DECISÃO, DT

As FIGURAS 92, 93, 94 e 95 apresentam os resultados do histórico de otimização para modelo DT nos experimentos I, II, III e IV, respectivamente. Diferente de outros modelos, não existe um comportamento claro dos experimentos. As diferentes configurações produziram resultados que, na média, divergiram em aproximadamente 0,05 de F1-score em todos os experimentos.

A FIGURA 96 apresenta o impacto no resultado e no tempo de cada hiperparâmetro estudado em cada experimento. No quesito impacto de resultado durante

FIGURA 79 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO MLP NO EXPERIMENTO III



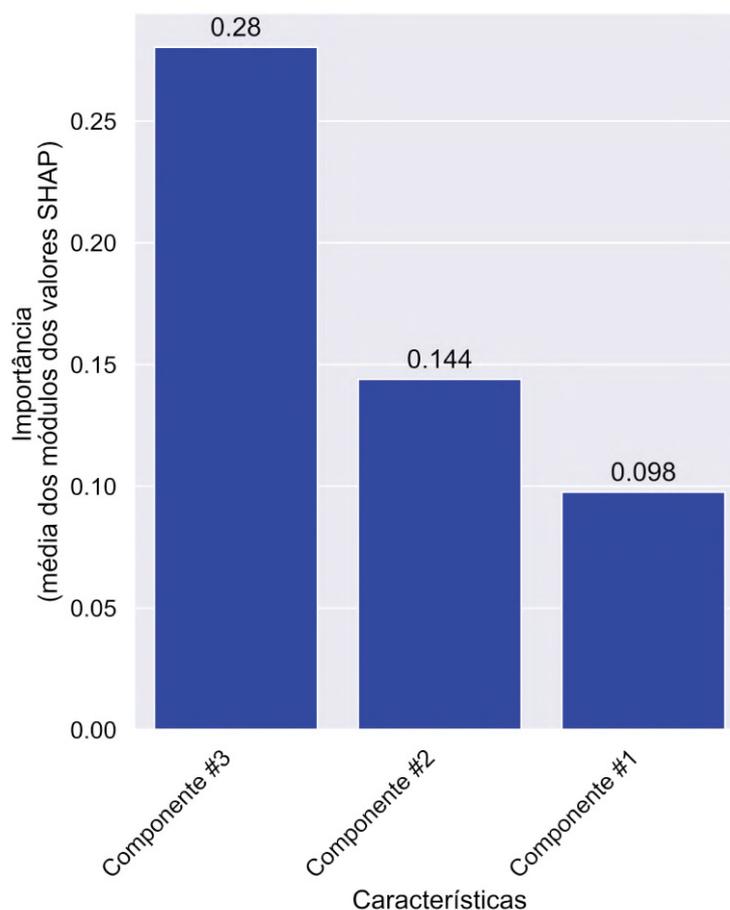
FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo MLP no experimento III.

o treinamento, o hiperparâmetro profundidade máxima foi que mais impactou. No experimento com uso de t-SNE, o hiperparâmetro critério de divisão também teve um impacto significativo. No quesito tempo, a estratégia de divisão foi o hiperparâmetro que mais impactou o tempo de otimização para os experimentos sem redução de dimensionalidade, com uso de PCA e com uso do t-SNE. Para experimento com UMAP, a profundidade máxima foi o hiperparâmetro que mais impactou o tempo.

Em relação a explicabilidade do modelo DT e a importância média do quanto cada característica ou componente no resultado da base de teste, as FIGURAS 97, 98, 99 e 100 mostram estes gráficos para cada um dos experimentos. No experimento sem redução de dimensionalidade (FIGURA 97), a característica que mais explicou o resultado foi a VLF. Enquanto que no experimento do PCA (FIGURA 98), as componentes 2 e 3 foram as que mais contribuíram. Para o experimento do t-SNE (FIGURA 99), a componente 2 teve maior impacto, enquanto que no experimento com UMAP (FIGURA 100), foi a componente 3, com as demais componentes pouco contribuindo

FIGURA 80 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO MLP NO EXPERIMENTO IV



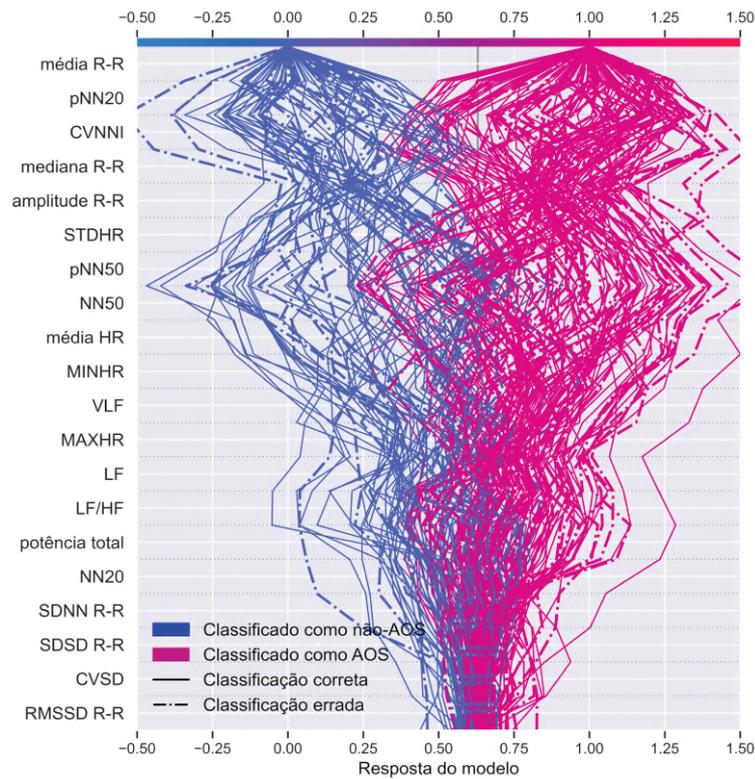
FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo MLP no experimento IV.

para a explicabilidade.

A FIGURA 101 mostra o processo decisório das 200 amostras para experimento sem redução de dimensionalidade. Na FIGURA, é possível observar que com a exceção das características do topo da FIGURA, VLF, pNN20, média HR e SDNN R-R, as demais contribuíram, em média, em pesos iguais para as linhas de predição. Todavia, alguns valores de RMSSD R-R, mediana R-R, média R-R parecem ter induzido a resposta do modelo para classe errada. Na FIGURA 102, que mostra o mesmo gráfico para experimento com PCA, é possível ver que determinados valores das componentes que, na média, menos contribuíram (componentes 9, 4 e 10) direcionam a resposta do modelo para um evento de AOS. Uma análise mais aprofundada deste comportamento pode ajudar na melhoria do desempenho quando PCA é aplicado. Também é possível observar que valores da componente 2 e 3 ajudaram o modelo a classificar incorretamente um evento de AOS. A FIGURA 103, que ilustra o gráfico para experimento t-SNE, as linhas de predição tiveram um comportamento muito mais homogêneo. Na

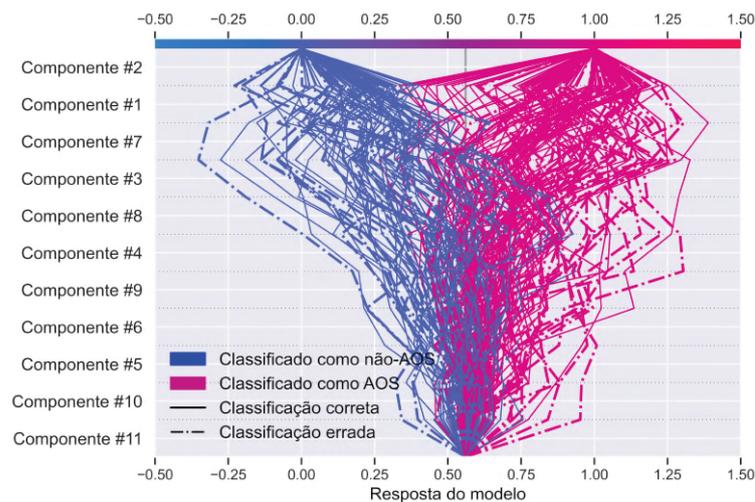
FIGURA 81 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO MLP NO EXPERIMENTO I



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo MLP no experimento I.

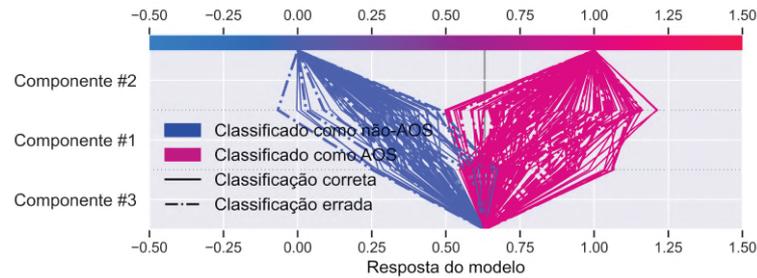
FIGURA 82 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO MLP NO EXPERIMENTO II



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo AdaBoost no experimento II.

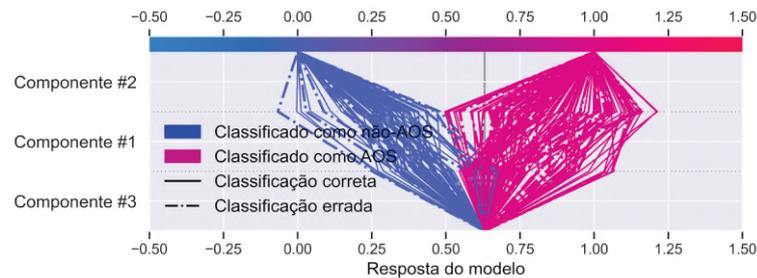
FIGURA 83 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO MLP NO EXPERIMENTO III



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo AdaBoost no experimento III.

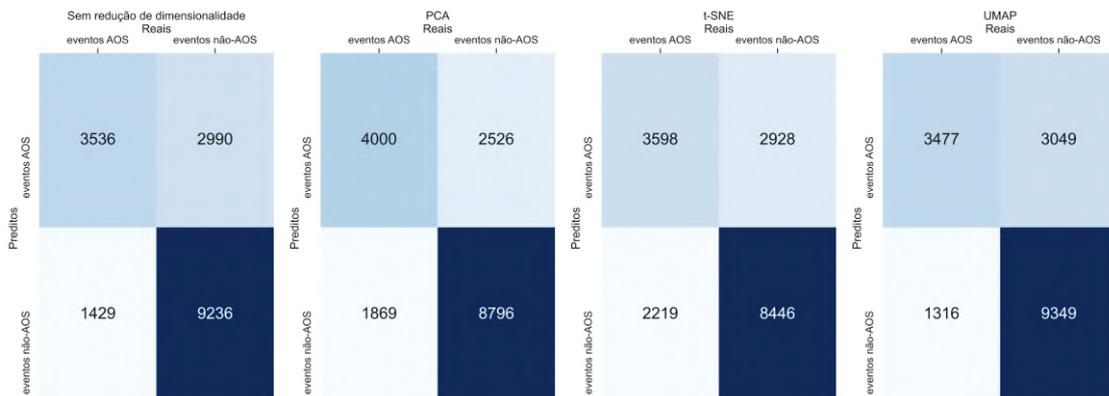
FIGURA 84 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO MLP NO EXPERIMENTO IV



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo AdaBoost no experimento IV.

FIGURA 85 – MATRIZ DE CONFUSÃO DO MODELO MLP

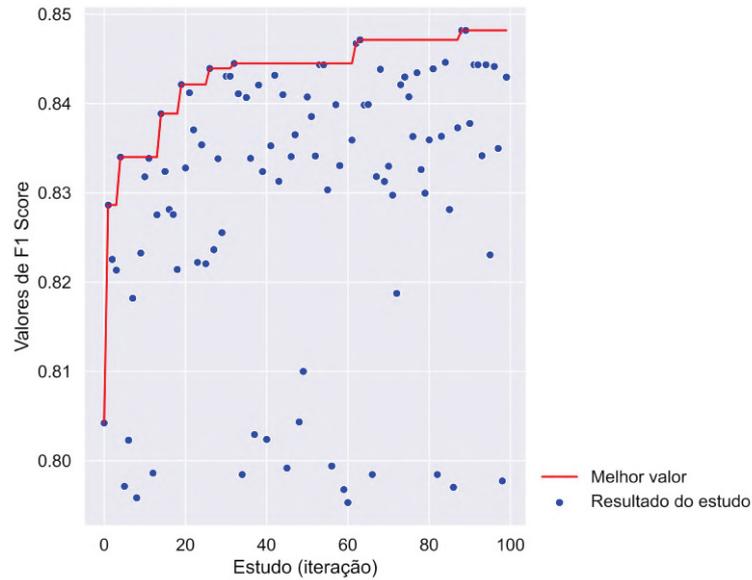


FONTE: Próprio autor

LEGENDA: Matriz de confusão para o modelo MLP em cada um dos experimentos.

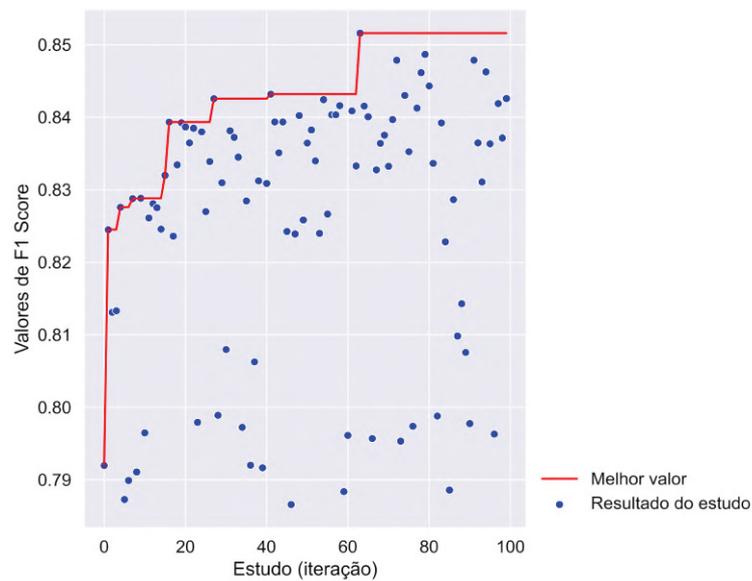
FIGURA FIGURA 104, do experimento com UMAP, é possível observar que todas as componentes contribuem para o modelo classificar um evento como AOS, mas para a classificação de um evento de não-AOS, componente 3 tem um peso maior nas predições.

A FIGURA 105 apresenta todas as matrizes de confusão dos experimentos

FIGURA 86 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO *BAGGING* NO EXPERIMENTO I

FONTE: Próprio autor

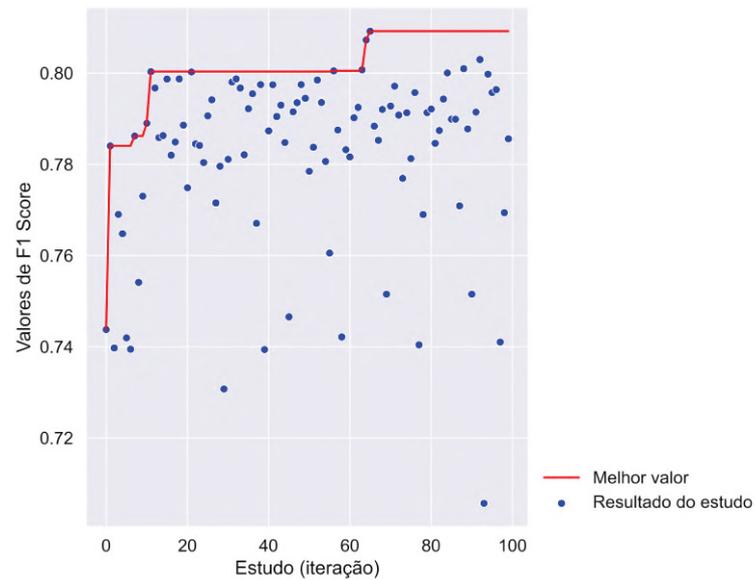
LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de *BAGGING* no experimento I.

FIGURA 87 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO *BAGGING* NO EXPERIMENTO II

FONTE: Próprio autor

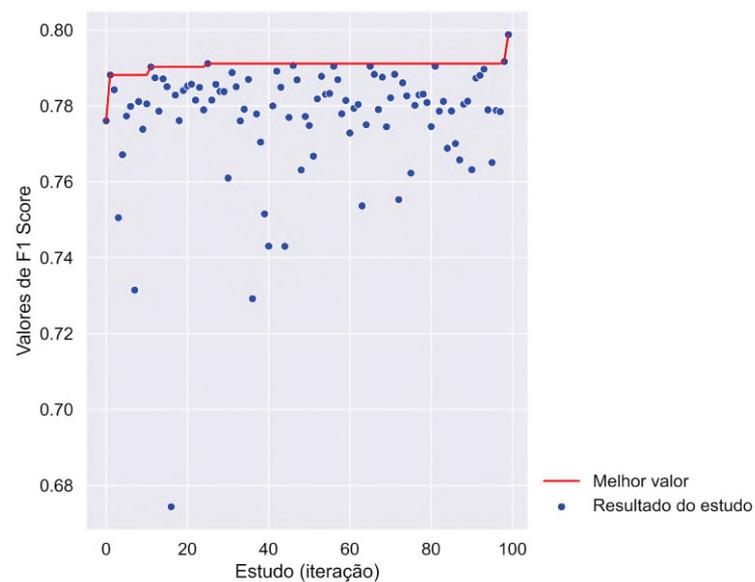
LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de *BAGGING* no experimento II.

para modelo DT. Analisando a FIGURA, pode-se observar que o experimento UMAP obteve maior acuracidade nos eventos de não-AOS reais, mas que teve um maior número de falsos negativos. Do experimento sem redução de dimensionalidade, para o experimento com PCA para o experimento com aplicação do t-SNE, a DT errou mais a classificação de eventos de AOS e de não-AOS.

FIGURA 88 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO *BAGGING* NO EXPERIMENTO III

FONTE: Próprio autor

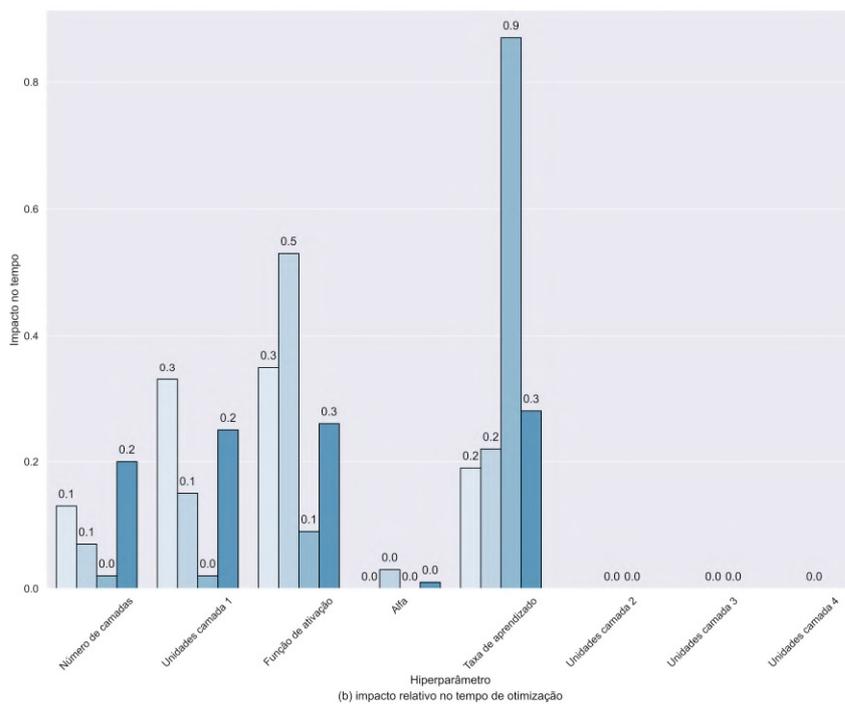
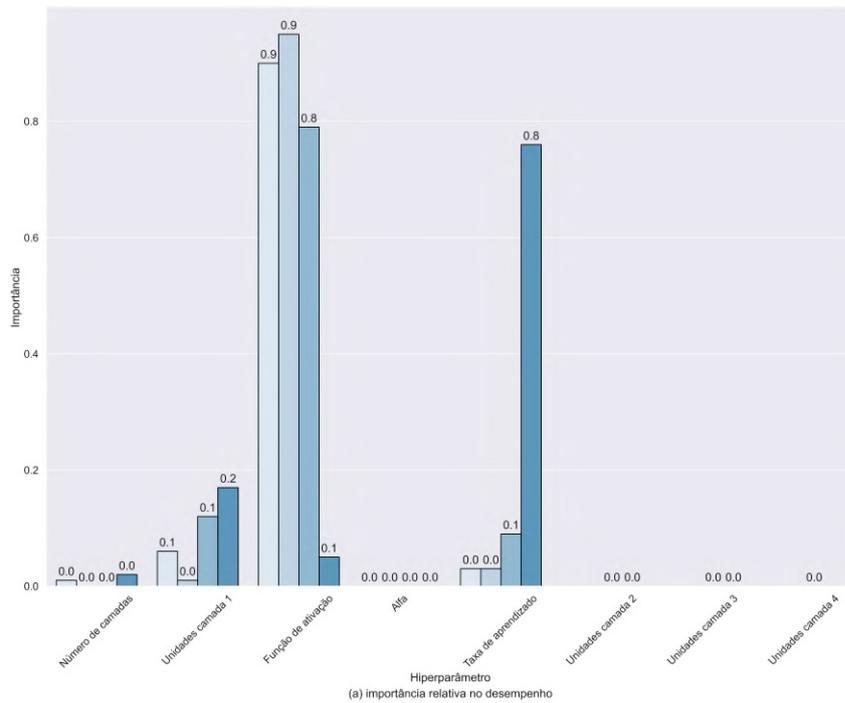
LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de *BAGGING* no experimento III.

FIGURA 89 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO *BAGGING* NO EXPERIMENTO IV

FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de *BAGGING* no experimento IV.

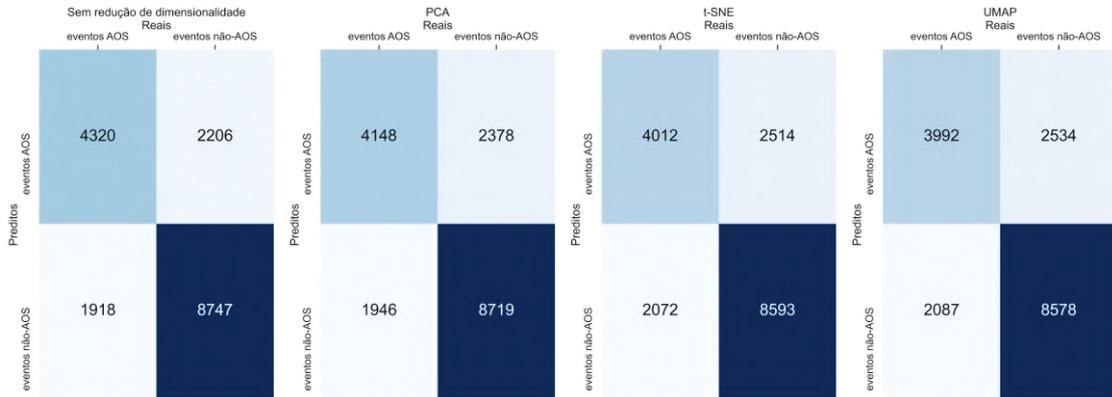
FIGURA 90 – IMPACTOS DOS HIPERPARÂMETROS NA OTIMIZAÇÃO DO MODELO *BAGGING*



Experimentos  
 □ sem redução de dimensionalidade  
 □ PCA  
 □ t-SNE  
 □ UMAP

FONTE: Próprio autor  
 LEGENDA: Impactos dos hiperparâmetros no desempenho e no tempo de otimização para modelo *bagging*.

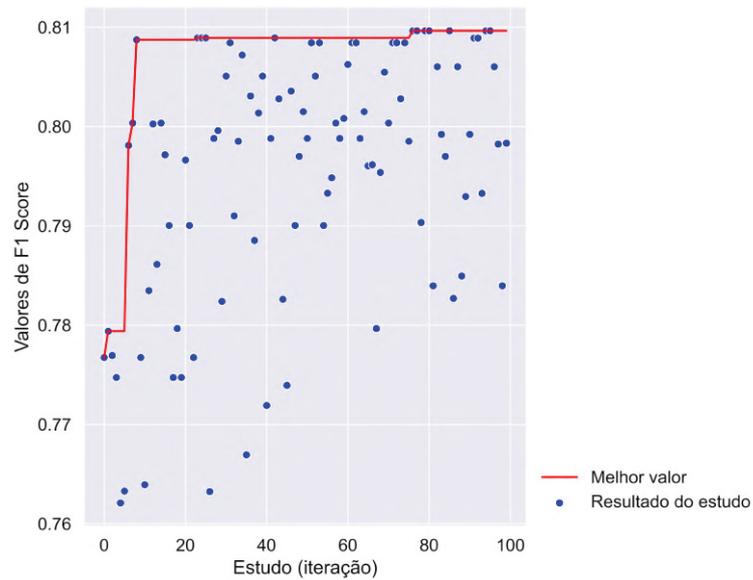
FIGURA 91 – MATRIZ DE CONFUSÃO DO MODELO *BAGGING*



FONTE: Próprio autor

LEGENDA: Matriz de confusão para o modelo *bagging* em cada um dos experimentos.

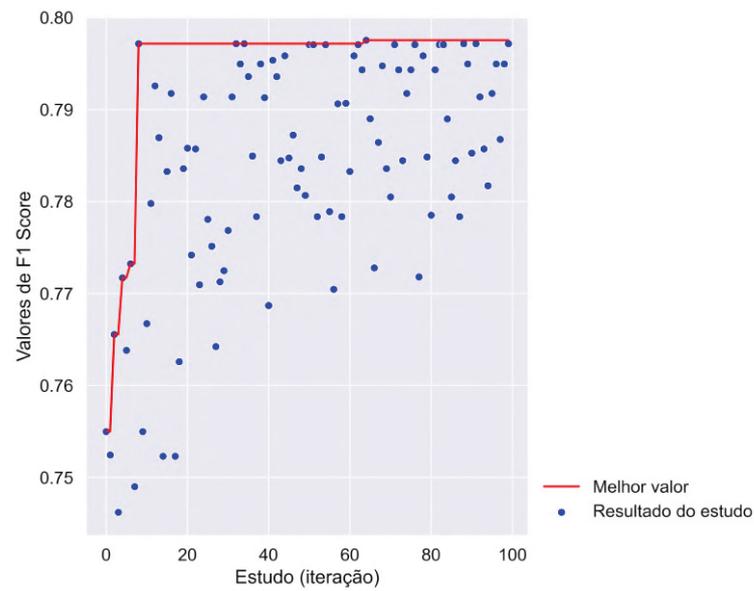
FIGURA 92 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO DT NO EXPERIMENTO I



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de DT no experimento I.

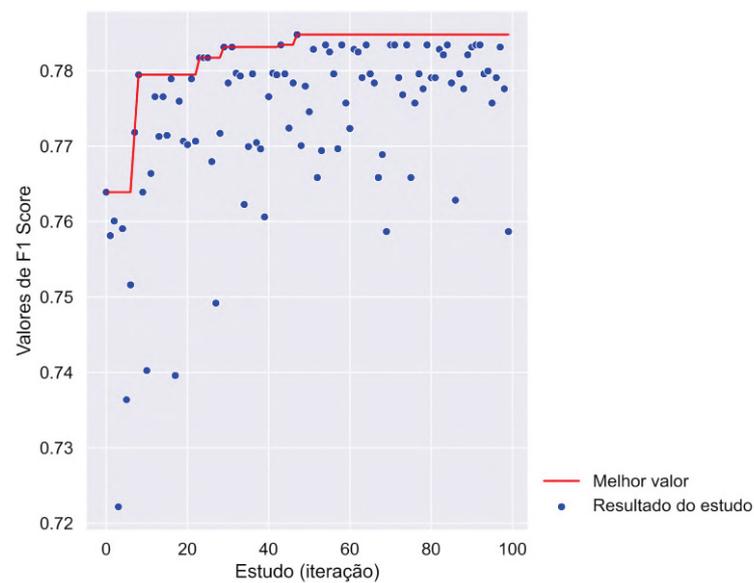
FIGURA 93 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO DT NO EXPERIMENTO II



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de DT no experimento II.

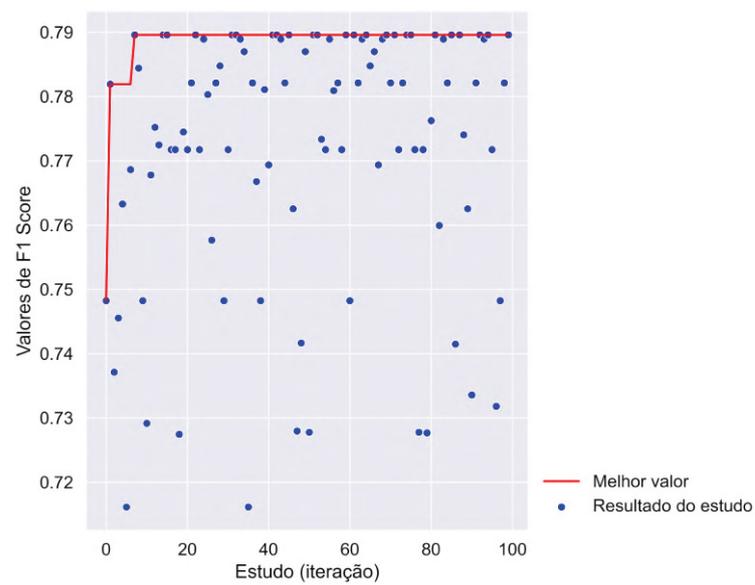
FIGURA 94 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO DT NO EXPERIMENTO III



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de DT no experimento III.

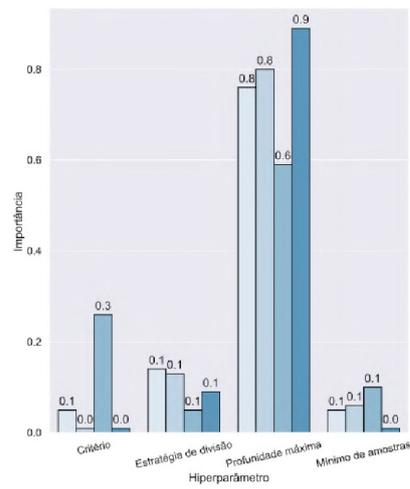
FIGURA 95 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO DT NO EXPERIMENTO IV



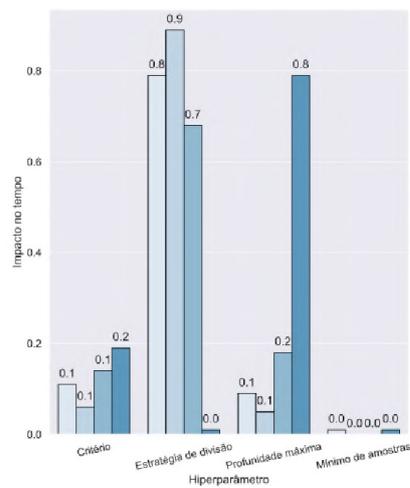
FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de DT no experimento IV.

FIGURA 96 – IMPACTOS DOS HIPERPARÂMETROS NA OTIMIZAÇÃO DO MODELO DT



(a) importância relativa no desempenho

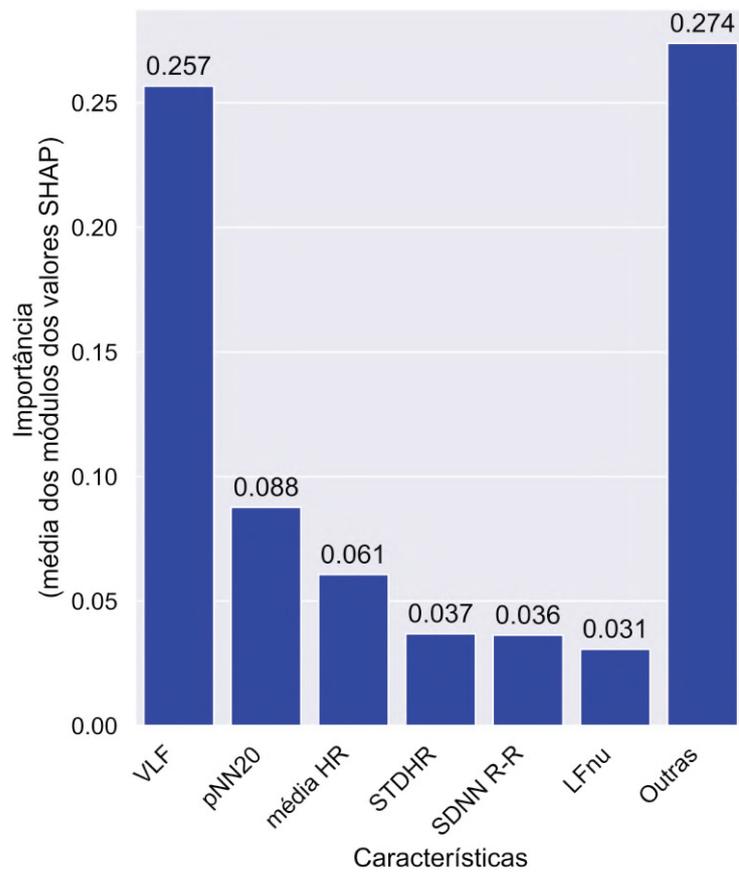


(b) impacto relativo no tempo de otimização

Experimentos  
 sem redução de dimensionalidade  
 PCA  
 t-SNE  
 UMAP

FONTE: Próprio autor  
 LEGENDA: Impactos dos hiperparâmetros no desempenho e no tempo de otimização para modelo DT.

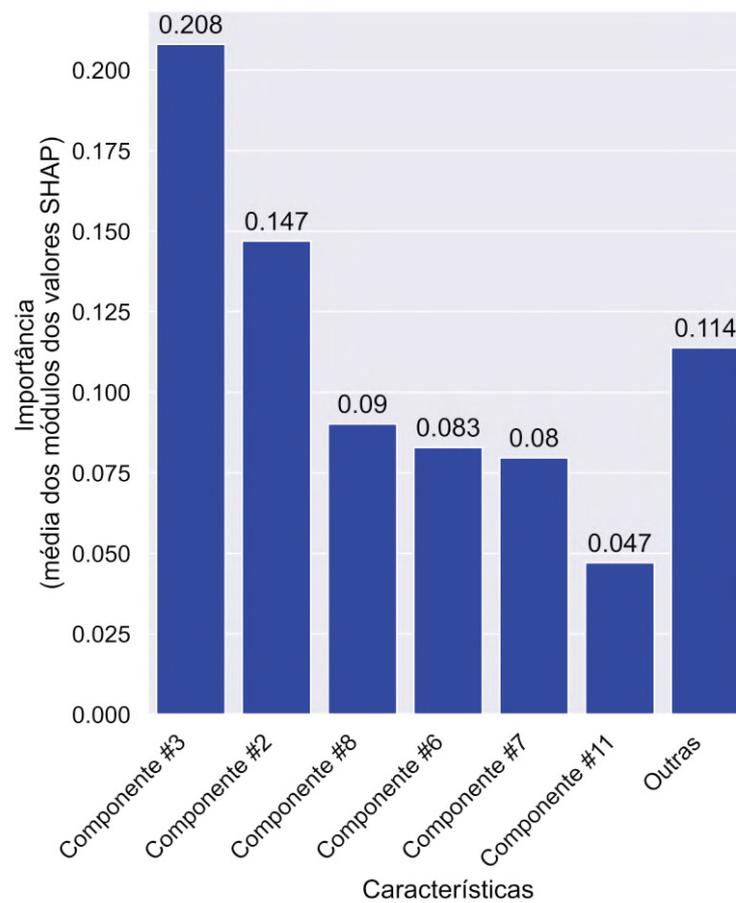
FIGURA 97 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO DT NO EXPERIMENTO I



FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo DT no experimento I.

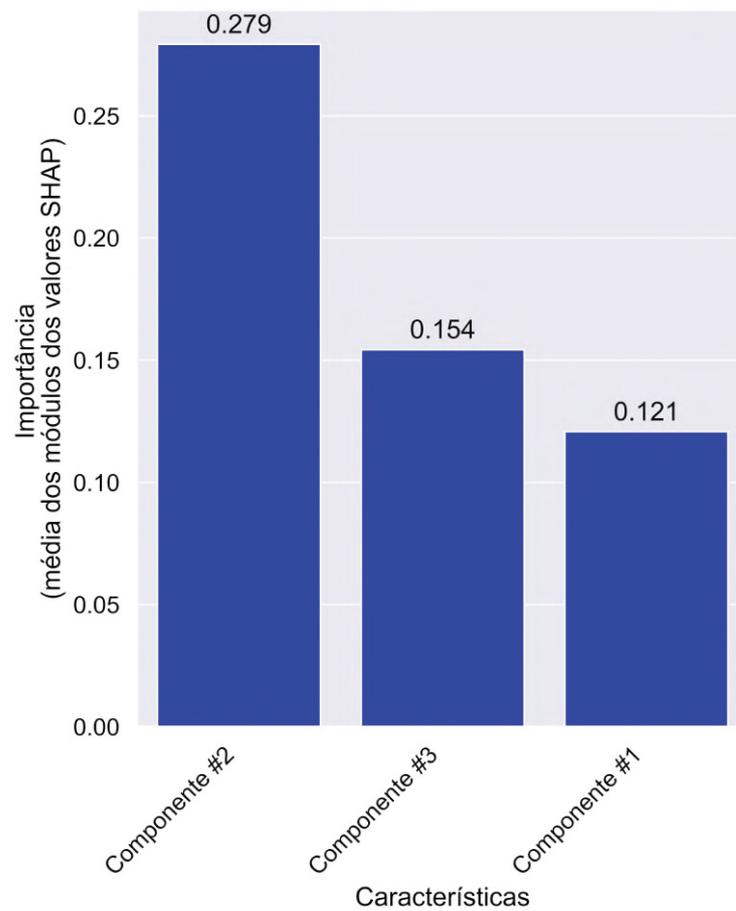
FIGURA 98 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO DT NO EXPERIMENTO II



FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo DT no experimento II.

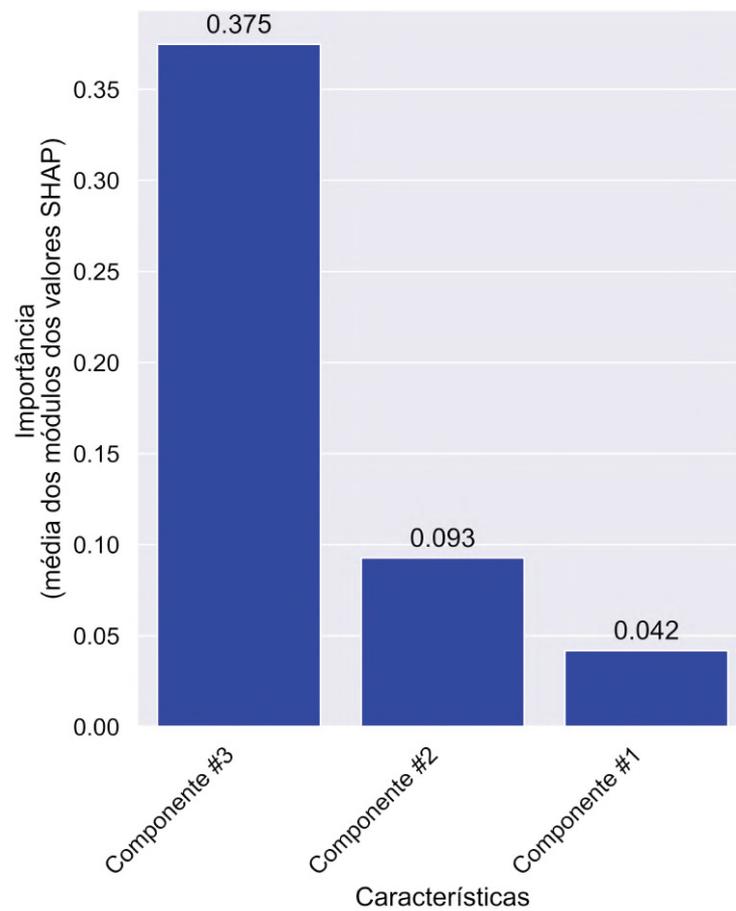
FIGURA 99 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO DT NO EXPERIMENTO III



FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo DT no experimento III.

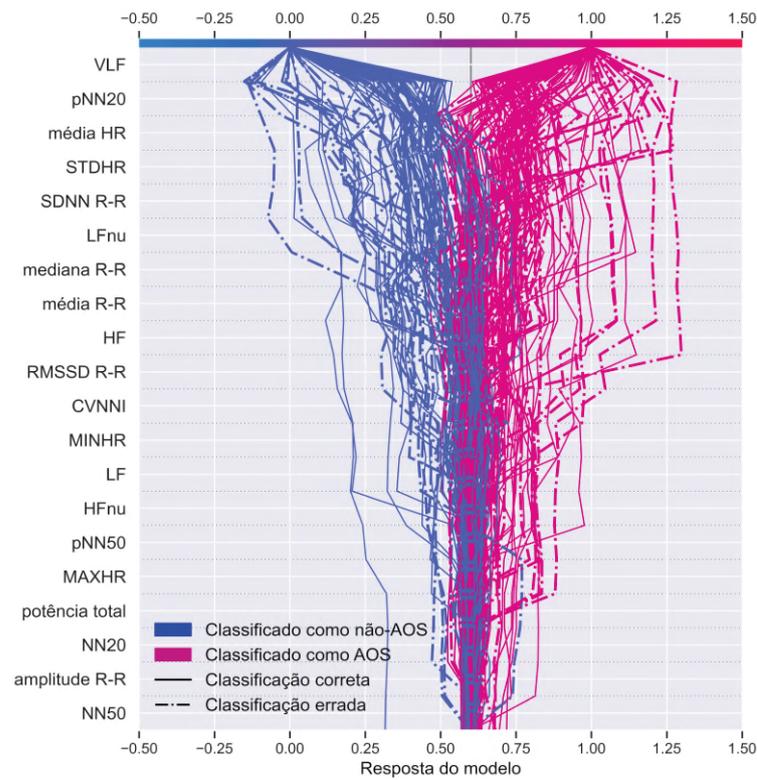
FIGURA 100 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO DT NO EXPERIMENTO IV



FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo DT no experimento IV.

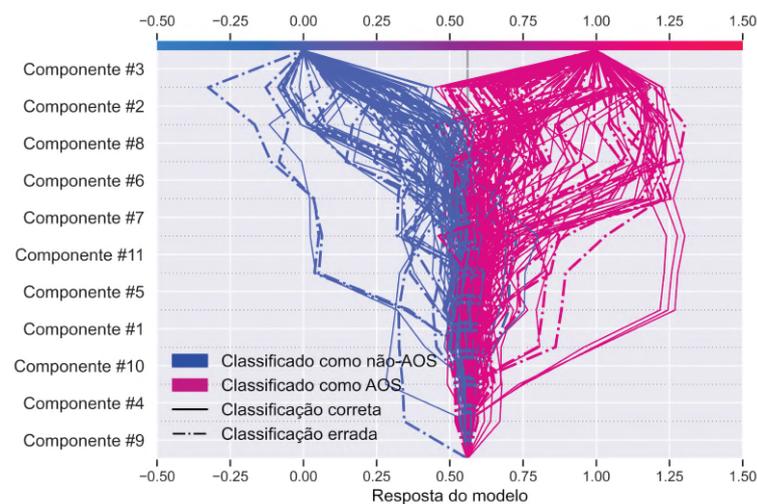
FIGURA 101 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO DT NO EXPERIMENTO I



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo DT no experimento I.

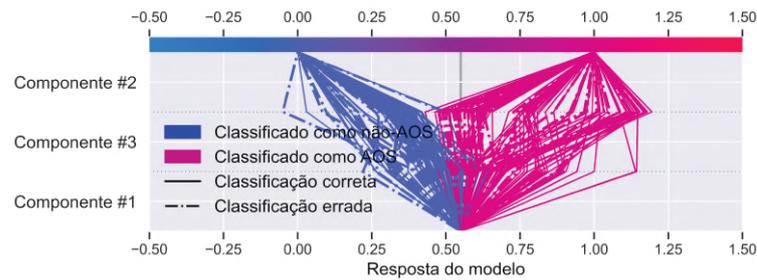
FIGURA 102 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO DT NO EXPERIMENTO II



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo DT no experimento II.

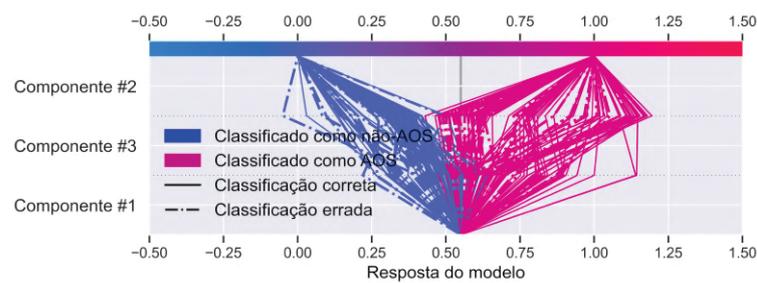
FIGURA 103 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO DT NO EXPERIMENTO III



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo DT no experimento III.

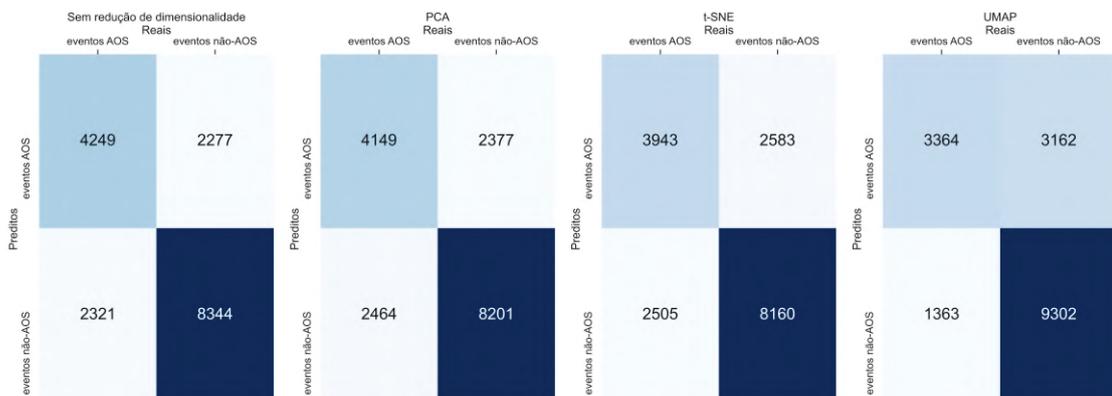
FIGURA 104 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO DT NO EXPERIMENTO IV



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo DT no experimento IV.

FIGURA 105 – MATRIZ DE CONFUSÃO DO MODELO DT



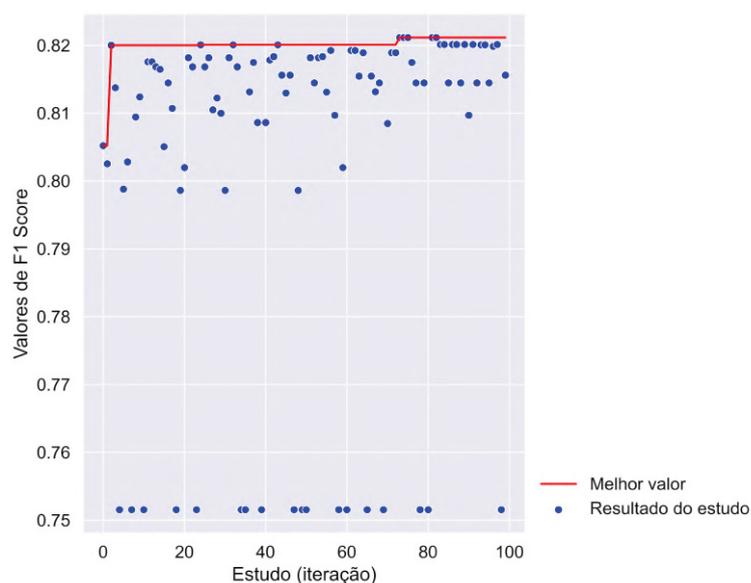
FONTE: Próprio autor

LEGENDA: Matriz de confusão para o modelo DT em cada um dos experimentos.

## DEMAIS RESULTADOS DO MODELO *ENSEMBLE* POR VOTO MAJORITÁRIO, VM

As FIGURAS 106, 107, 108 e 109 mostram os históricos de otimização para modelo VM nos experimentos I, II, III e IV respectivamente. Analisando as FIGURAS, é possível observar que as configurações testadas, em todos os experimentos, apresentaram duas faixas de resposta, com experimento III (FIGURA 108) apresentado uma maior variabilidade. No experimento I, FIGURA 106, uma faixa de resultados apresentou resultados entre 0,80 e 0,82 enquanto a segunda faixa de resultados ficou em aproximadamente 0,75. Na 107, a primeira faixa ficou entre cerca de 0,78 e 0,80, enquanto a segunda faixa ficou próxima de 0,75. Finalmente, para experimento IV, 109, a primeira faixa ficou entre 0,78 e 0,79 e a segunda faixa com valores próximo de 0,75. Isto pode ter acontecido, por se tratar de um modelo de *ensemble*, algum tipo de configuração com um tipo de modelo pode ter melhorado ou piorado os resultados. As próximas análises podem ajudar a analisar esta hipótese.

FIGURA 106 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO VM NO EXPERIMENTO I

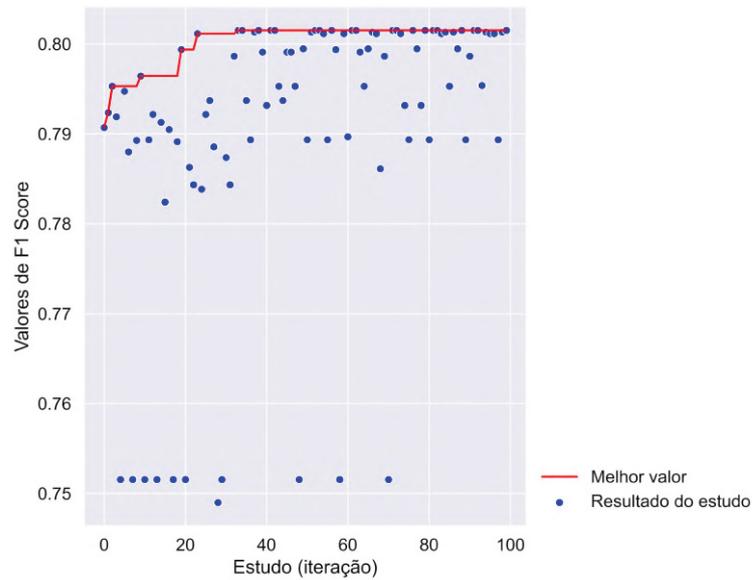


FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de VM no experimento I.

Diferentemente de como foi apresentado nos outros modelos, devido a grande quantidade de hiperparâmetros, os resultados sobre impacto no tempo e resultado dos hiperparâmetros são apresentados nas TABELAS 62, 63, 64 e 65 para os experimentos sem redução de dimensionalidade, com aplicação de PCA, t-SNE e UMAP respectivamente. O hiperparâmetro número total de estimadores foi o que mais teve impacto no tempo de processamento em todos os experimentos. A importância variou de acordo com os experimentos, mas os experimentos com t-SNE e UMAP (TABELAS 64 e 65, respectivamente), com menor dimensão, a melhor configuração encontrada

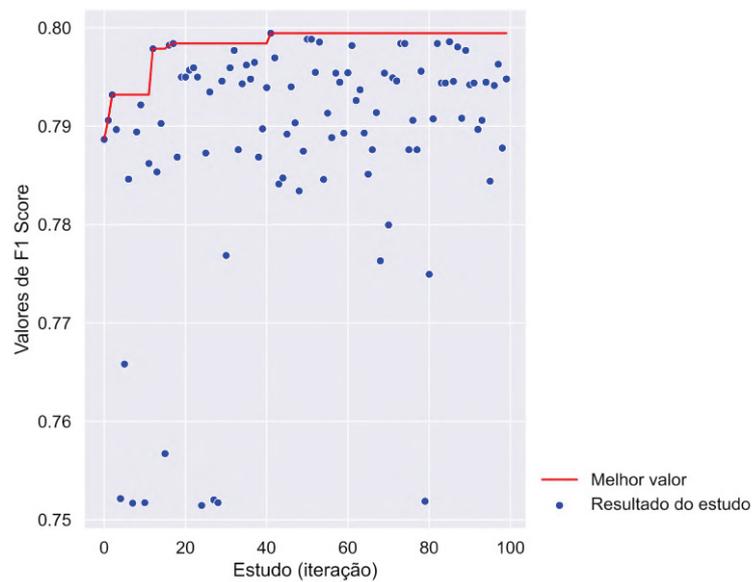
FIGURA 107 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO VM NO EXPERIMENTO II



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de VM no experimento II.

FIGURA 108 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO VM NO EXPERIMENTO III



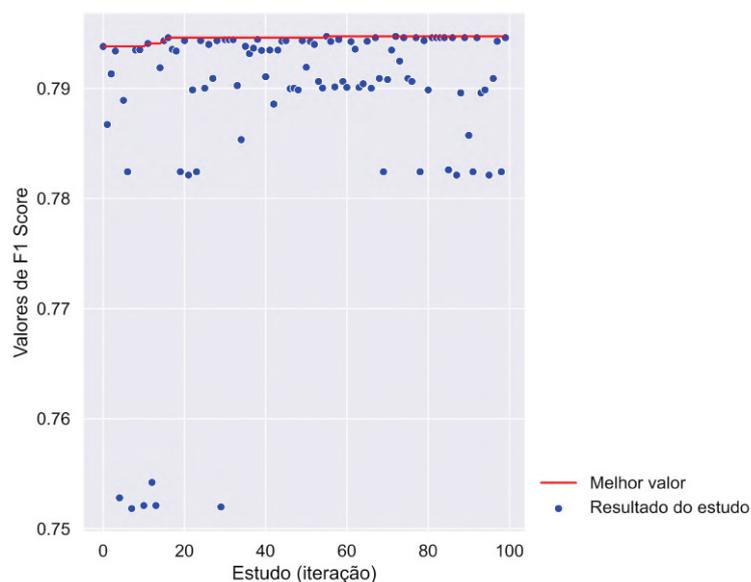
FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de VM no experimento III.

possui mais estimadores do que nos experimentos de maior dimensão (TABELAS 62 e 63). Isto pode ter acontecido, pois com a redução para três dimensões, o modelo precisou agregar mais classificadores para capturar mais padrões nos dados.

Não foram realizadas análises com valores SHAP com o modelo VM. A FIGURA 110 apresenta as matrizes de confusão do modelo VM para os experimentos

FIGURA 109 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO VM NO EXPERIMENTO IV



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de VM no experimento IV.

TABELA 62 – IMPACTO RELATIVO NO RESULTADO E NO TEMPO DE CADA HIPERPARÂMETRO DO MODELO VM NO EXPERIMENTO I

Hiperparâmetro	Valor	Importância relativa	Tempo relativo
N° de estimadores	3,00	0,10	0,99
Profundidade arv. 1	9,00	0,04	$1,00^{-6}$
C SVM 1	93756,31	0,08	$1,00^{-6}$
Gama SVM 1	48879,61	0,08	$1,00^{-6}$
Profundidade arv. 2	7,00	0,06	$1,00^{-6}$
C SVM 2	7896,60	0,03	$1,00^{-6}$
Gama SVM 2	7873,04	0,04	$1,00^{-6}$
Peso arv. 1	0,38	0,36	$1,00^{-6}$
Peso SVM 1	0,85	0,12	$1,00^{-6}$
Peso arv. 2	0,91	0,05	$1,00^{-6}$
Peso SVM 2	0,77	0,03	$1,00^{-6}$

FONTE: Elaborada pelo autor

realizados. Na FIGURA, é possível observar que no experimento UMAP, o modelo foi capaz de classificar corretamente mais eventos de não-AOS, no entanto, a quantidade de falsos negativos também cresceu em relação aos demais experimentos. Isso pode demonstrar que a redução de dimensionalidade com UMAP fez com que o modelo VM tendesse a classificar mais eventos como não-AOS. A quantidade de eventos AOS classificados corretamente foi maior no experimento sem redução de dimensionalidade, seguido pelo experimento t-SNE, pelo PCA e por fim UMAP.

TABELA 63 – IMPACTO RELATIVO NO RESULTADO E NO TEMPO DE CADA HIPERPARÂMETRO DO MODELO VM NO EXPERIMENTO II

Hiperparâmetro	Valor	Importância relativa	Tempo relativo
N° de estimadores	3,00	0,10	0,99
Profundidade arv. 1	9,00	0,04	1,00 <sup>-6</sup>
C SVM 1	93756,31	0,08	1,00 <sup>-6</sup>
Gama SVM 1	48879,61	0,08	1,00 <sup>-6</sup>
Profundidade arv. 2	7,00	0,06	1,00 <sup>-6</sup>
C SVM 2	7896,60	0,03	1,00 <sup>-6</sup>
Gama SVM 2	7873,04	0,04	1,00 <sup>-6</sup>
Profundidade arv. 3	6,00	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
C SVM 3	51232,21	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
Gama SVM 3	44921,14	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
Peso arv. 1	0,38	0,36	1,00 <sup>-6</sup>
Peso SVM 1	0,85	0,12	1,00 <sup>-6</sup>
Peso arv. 2	0,91	0,05	1,00 <sup>-6</sup>
Peso SVM 2	0,77	0,03	1,00 <sup>-6</sup>
Peso arv. 3	0,68	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
Peso SVM 3	0,14	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>

FONTE: Elaborada pelo autor

TABELA 64 – IMPACTO RELATIVO NO RESULTADO E NO TEMPO DE CADA HIPERPARÂMETRO DO MODELO VM NO EXPERIMENTO III

Hiperparâmetro	Valor	Importância relativa	Tempo relativo
N° de estimadores	4,00	0,10	0,99
Profundidade arv. 1	9,00	0,06	1,00 <sup>-6</sup>
C SVM 1	5431,55	0,10	1,00 <sup>-6</sup>
Gama SVM 1	6310,77	0,10	1,00 <sup>-6</sup>
Profundidade arv. 2	8,00	0,06	1,00 <sup>-6</sup>
C SVM 2	53348,39	0,10	1,00 <sup>-6</sup>
Gama SVM 2	82617,07	0,15	1,00 <sup>-6</sup>
Profundidade arv. 3	3,00	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
C SVM 3	20019,56	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
Gama SVM 3	15519,74	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
Profundidade arv. 4	6,00	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
C SVM 4	10967,17	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
Gama SVM 4	45849,56	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
Peso arv. 1	0,56	0,15	1,00 <sup>-6</sup>
Peso SVM 1	0,50	0,09	1,00 <sup>-6</sup>
Peso arv. 2	0,82	0,03	1,00 <sup>-6</sup>
Peso SVM 2	0,05	0,05	1,00 <sup>-6</sup>
Peso arv. 3	0,36	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
Peso SVM 3	0,14	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
Peso arv. 4	0,65	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
Peso SVM 4	0,02	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>

FONTE: Elaborada pelo autor

DEMAIS RESULTADOS DO MODELO *LIGHT GRADIENT BOOSTING*, LIGHTGBM

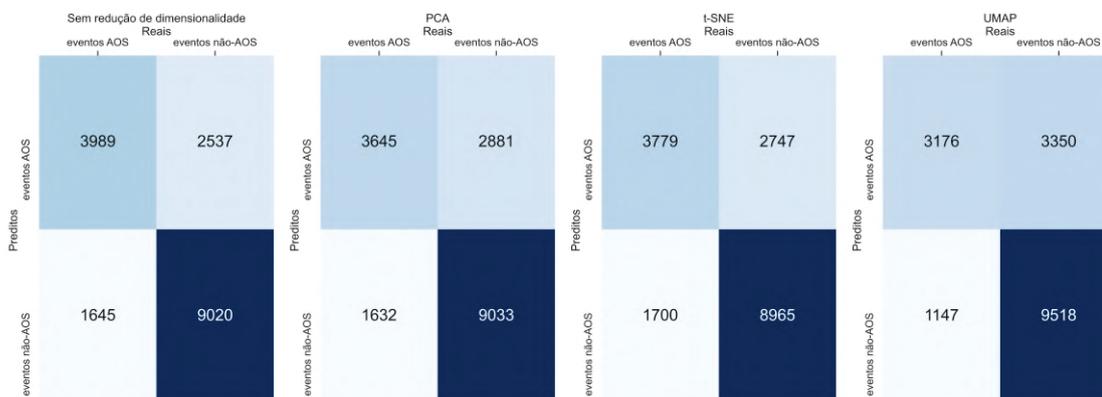
As FIGURAS 111, 112, 113 e 114 mostram o histórico de otimização para os experimentos com LightGBM. Na FIGURA 111, histórico de otimização do experimento sem redução de dimensionalidade, os resultados divergeriam em cerca de 0,03, com uma grande quantidade de resultados próximas a melhor configuração obtida. O comportamento no experimento PCA, FIGURA 112, foi semelhante, com uma amplitude de

TABELA 65 – IMPACTO RELATIVO NO RESULTADO E NO TEMPO DE CADA HIPERPARÂMETRO DO MODELO VM NO EXPERIMENTO IV

Hiperparâmetro	Valor	Importância relativa	Tempo relativo
N° de estimadores	4,00	1,00 <sup>-6</sup>	0,99
Profundidade arv. 1	4,00	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
C SVM 1	85624,13	0,08	1,00 <sup>-6</sup>
Gama SVM 1	54775,21	0,15	1,00 <sup>-6</sup>
Profundidade arv. 2	7,00	0,04	1,00 <sup>-6</sup>
C SVM 2	58861,08	0,04	1,00 <sup>-6</sup>
Gama SVM 2	30564,88	0,04	1,00 <sup>-6</sup>
Profundidade arv. 3	3,00	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
C SVM 3	37967,86	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
Gama SVM 3	28436,17	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
Profundidade arv. 4	6,00	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
C SVM 4	24696,43	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
Gama SVM 4	19133,43	0	1,00 <sup>-6</sup>
Peso arv. 1	0,77	0,33	1,00 <sup>-6</sup>
Peso SVM 1	0,58	0,16	1,00 <sup>-6</sup>
Peso arv. 2	0,83	0,02	1,00 <sup>-6</sup>
Peso SVM 2	0,32	0,13	1,00 <sup>-6</sup>
Peso arv. 3	0,94	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
Peso SVM 3	0,46	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
Peso arv. 4	0,45	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>
Peso SVM 4	0,55	1,00 <sup>-6</sup>	1,00 <sup>-6</sup>

FONTE: Elaborada pelo autor

FIGURA 110 – MATRIZ DE CONFUSÃO DO MODELO VM



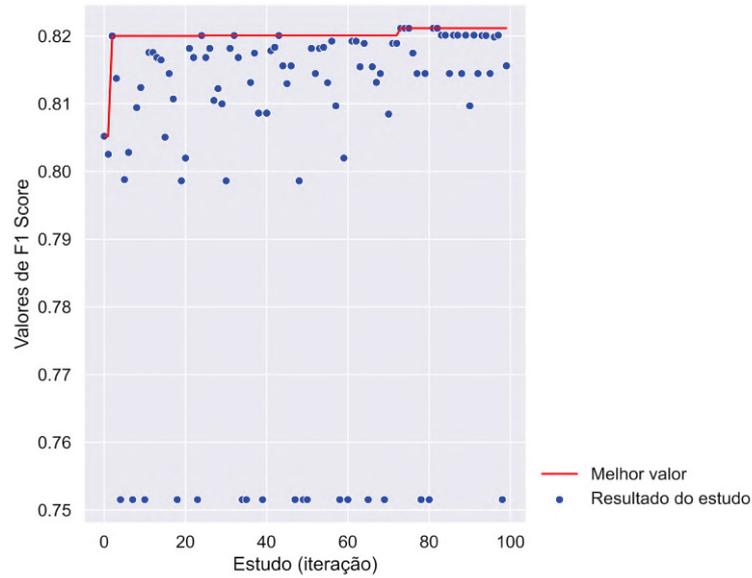
FONTE: Próprio autor

LEGENDA: Matriz de confusão para o modelo VM em cada um dos experimentos.

variação maior, em cerca de 0,05. A amplitude de variação diminuiu em relação aos dois primeiros experimentos, tanto na FIGURA 113, do experimento do t-SNE, quanto na FIGURA 114, do experimento UMAP, em que esta variação ficou em aproximadamente 0,02.

Como no caso do modelo LightGBM muitos hiperparâmetros foram estudados, os impactos de cada hiperparâmetro no resultado e no tempo de treinamento são apresentados nas TABELAS 66, 67, 68 e 69, respectivamente para os experimentos I, II, III e IV. Os dois hiperparâmetros que mais impactaram no tempo foram número de estimadores e a profundidade máxima das árvores em todos os experimentos. Para o

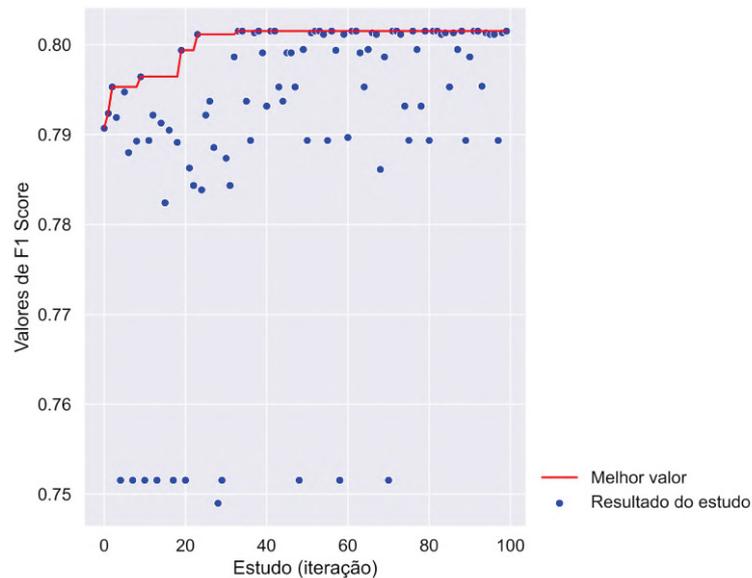
FIGURA 111 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO LIGHTGBM NO EXPERIMENTO I



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de LightGBM no experimento I.

FIGURA 112 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO LIGHTGBM NO EXPERIMENTO II

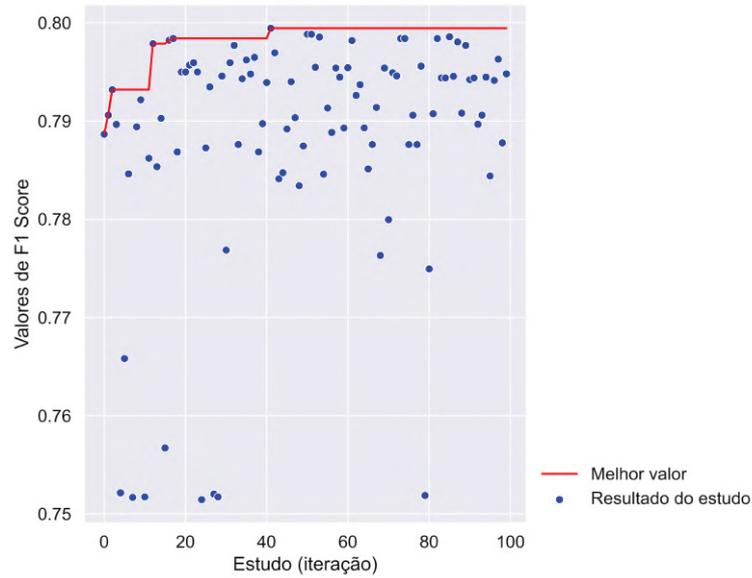


FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de LightGBM no experimento II.

experimentos com um espaço de característico maior, I e II (mostrados respectivamente nas TABELAS 66 e 67), o hiperparâmetro profundidade máxima foi o que mais impactou o resultado. Diferentemente do que aconteceu nos experimentos III e IV (TABELAS 68 e 69, respectivamente), com menor espaço de características, o hiperparâmetro

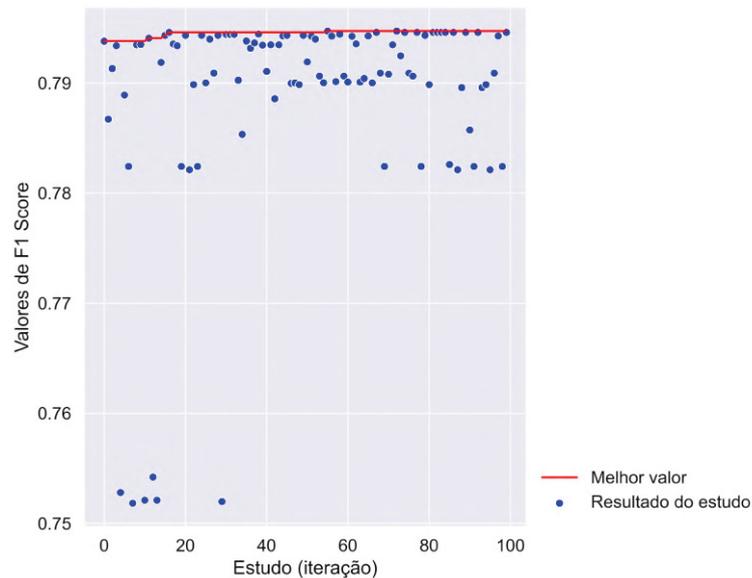
FIGURA 113 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO LIGHTGBM NO EXPERIMENTO III



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de LightGBM no experimento III.

FIGURA 114 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO LIGHTGBM NO EXPERIMENTO IV



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de LightGBM no experimento IV.

razão de amostras de coluna na base de treinamento. Isto pode ter acontecido devido justamente uma menor quantidade de colunas (espaço de características menor).

Devidos à restrições computacionais, não foram calculados valores SHAP para

TABELA 66 – IMPACTO RELATIVO NO RESULTADO E NO TEMPO DE CADA HIPERPARÂMETRO DO MODELO LIGHTGBM NO EXPERIMENTO I

Hiperparâmetro	Valor	Importância relativa	Tempo relativo
N° de folhas	236,00	0,05	0,04
Taxa de aprendizado	0,07	$1,00^{-6}$	0,03
Profundidade máxima	13,00	0,77	0,27
Soma mínima de pesos	14,00	$1,00^{-6}$	0,13
Razão de amostragem para treinamento	0,18	$1,00^{-6}$	$1,00^{-6}$
Razão de amostras de coluna	0,81	0,10	0,03
Regulador alfa	3,26	$1,00^{-6}$	0,03
Regulador lambda	5,41	0,04	0,02
N° de estimadores	158,00	0,03	0,45

FONTE: Elaborada pelo autor

TABELA 67 – IMPACTO RELATIVO NO RESULTADO E NO TEMPO DE CADA HIPERPARÂMETRO DO MODELO LIGHTGBM NO EXPERIMENTO II

Hiperparâmetro	Valor	Importância relativa	Tempo relativo
N° de folhas	115,00	0,06	0,14
Taxa de aprendizado	0,06	0,02	0,02
Profundidade máxima	13,00	0,64	0,41
Soma mínima de pesos	23,00	0,01	0,03
Razão de amostragem para treinamento	0,55	0,01	0,01
Razão de amostras de coluna	1,00	0,14	0,05
Regulador alfa	7,07	0,01	0,06
Regulador lambda	9,08	0,02	0,01
N° de estimadores	478,00	0,09	0,27

FONTE: Elaborada pelo autor

TABELA 68 – IMPACTO RELATIVO NO RESULTADO E NO TEMPO DE CADA HIPERPARÂMETRO DO MODELO LIGHTGBM NO EXPERIMENTO III

Hiperparâmetro	Valor	Importância relativa	Tempo relativo
N° de folhas	141,00	$1,00^{-6}$	0,06
Taxa de aprendizado	0,10	0,01	0,08
Profundidade máxima	15,00	0,01	0,36
Soma mínima de pesos	41,00	$1,00^{-6}$	0,02
Razão de amostragem para treinamento	0,47	0,01	0,01
Razão de amostras de coluna	0,84	0,91	0,02
Regulador alfa	4,44	0,01	0,06
Regulador lambda	2,70	0,03	0,01
N° de estimadores	107,00	$1,00^{-6}$	0,38

FONTE: Elaborada pelo autor

análise de explicabilidade para o modelo LightGBM. A FIGURA 115 apresenta as matrizes de confusão do modelo LightGBM. Na FIGURA, o experimento sem redução de dimensionalidade foi que melhor classificou corretamente eventos de AOS. Este acerto caiu nos outros três experimentos, com menor valor obtido no experimento do t-SNE. Quanto ao quesito de classificação de eventos de não-AOS, o melhor resultado foi obtido pelo experimento UMAP, seguido pelo experimento sem redução de dimensionalidade, PCA e, por último, t-SNE.

TABELA 69 – IMPACTO RELATIVO NO RESULTADO E NO TEMPO DE CADA HIPERPARÂMETRO DO MODELO LIGHTGBM NO EXPERIMENTO IV

Hiperparâmetro	Valor	Importância relativa	Tempo relativo
N° de folhas	123,00	0,01	0,02
Taxa de aprendizado	0,03	$1,00^{-6}$	0,13
Profundidade máxima	9,00	0,04	0,25
Soma mínima de pesos	6,00	0,02	0,05
Razão de amostragem para treinamento	0,45	$1,00^{-6}$	0,01
Razão de amostras de coluna	0,88	0,88	0,04
Regulador alfa	1,17	$1,00^{-6}$	0,01
Regulador lambda	4,21	0,02	0,17
N° de estimadores	90,00	0,02	0,32

FONTE: Elaborada pelo autor

FIGURA 115 – MATRIZ DE CONFUSÃO DO MODELO LIGHTGBM



FONTE: Próprio autor

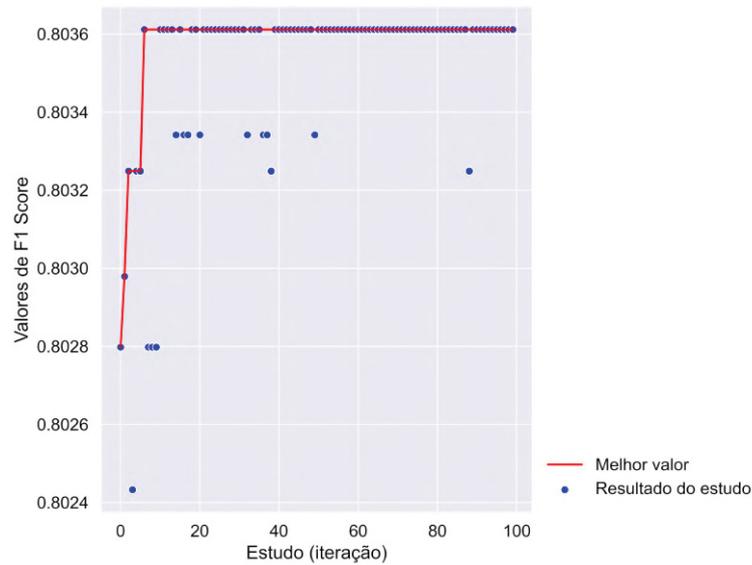
LEGENDA: Matriz de confusão para o modelo LightGBM em cada um dos experimentos.

## DEMAIS RESULTADOS DO MODELO REGRESSÃO LOGÍSTICA, LR

Nas FIGURAS 116, 116, 116 e 116. A FIGURA 116 mostra esta evolução para o experimento I. Como o modelo de LR possui uma quantidade pequena de hiperparâmetros (apenas três estudados), é possível observar faixas de resultados à medida que as configurações possíveis são testadas, que obtiveram resultados muito próximos, diferindo em cerca de 0,0012 de F-1 Score. Como a quantidade de configurações são reduzidas, é possível observar que o Optuna encontrou a melhor configuração e não conseguiu outra combinação que superasse este melhor valor. O mesmo comportamento aconteceu com a FIGURA 117, do experimento com PCA, e da FIGURA 119, do experimento com UMAP, com faixas de resultado ainda mais homogêneas. Para o experimento t-SNE mostrado na FIGURA 118, O Optuna não conseguiu encontrar combinações de hiperparâmetros que superassem o resulta de 0,74 de F1-score.

A FIGURA 120 analisa a importância e impacto no tempo relativo de cada hiperparâmetro em cada experimento. Na FIGURA, é possível observar que o hiperparâmetro C foi o que mais contribuiu para o desempenho na base de validação, enquanto

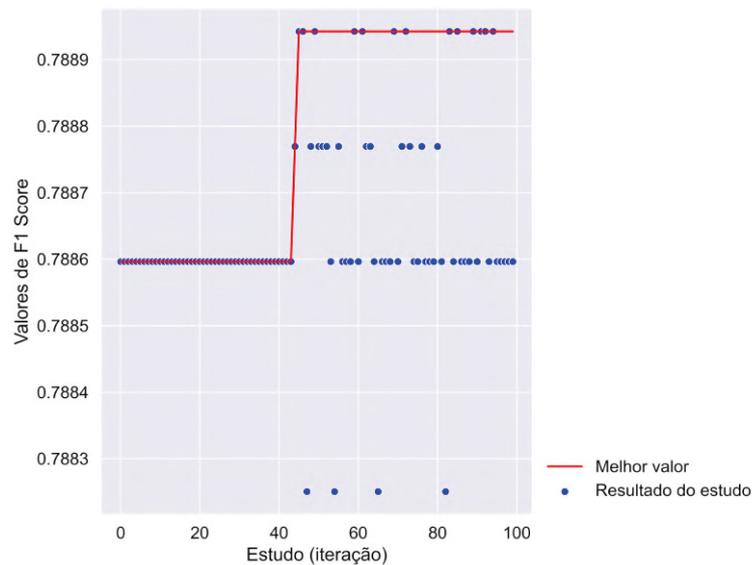
FIGURA 116 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO LR NO EXPERIMENTO I



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de LR no experimento I.

FIGURA 117 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO LR NO EXPERIMENTO II



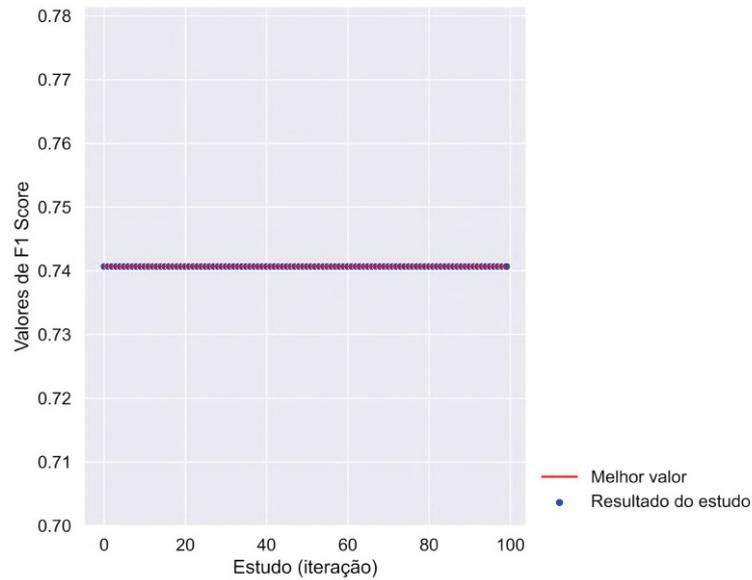
FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de LR no experimento II.

que o hiperparâmetro penalidade, foi o que mais impactou o tempo de otimização.

A FIGURA 121 mostra a importância média de cada característica na contribuição e explicação do resultado do modelo na base de teste n experimento sem redução de dimensionalidade. Houve um equilíbrio na importância de todas as características, sendo a característica média R-R e pNN20 as que mais impactaram o resultado do

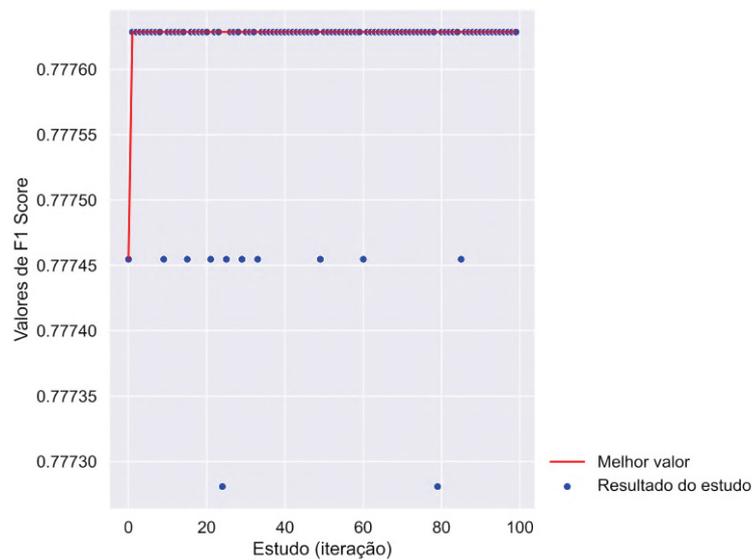
FIGURA 118 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO LR NO EXPERIMENTO III



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de LR no experimento III.

FIGURA 119 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO LR NO EXPERIMENTO IV

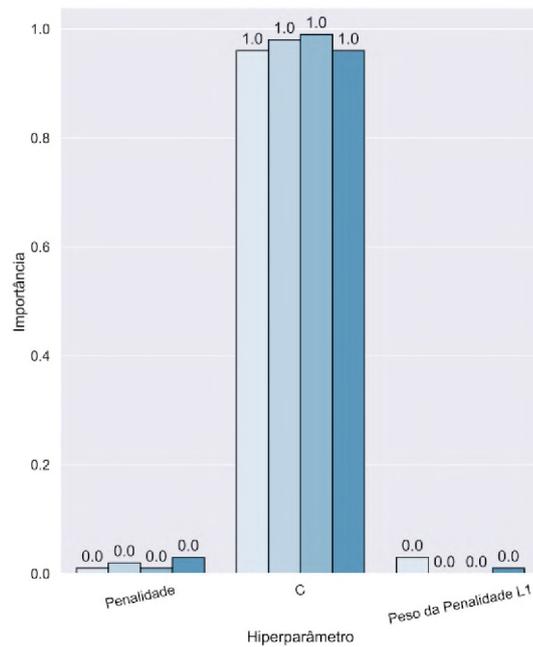


FONTE: Próprio autor

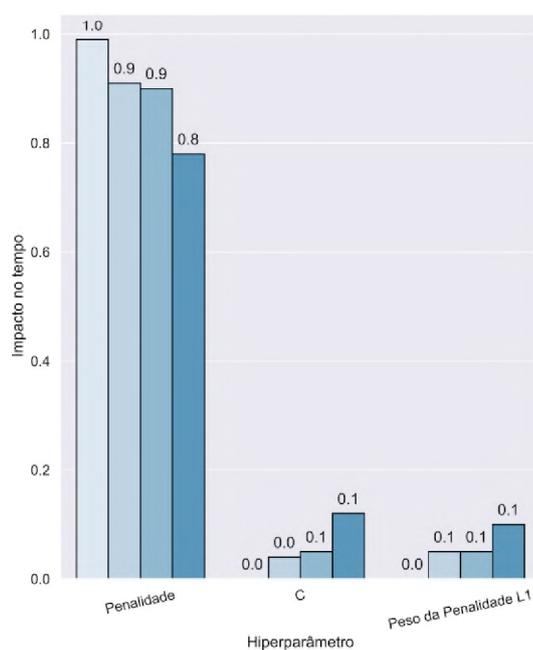
LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de LR no experimento IV.

modelo, ainda que não de forma dominante. Na FIGURA 122 a mesma análise é feita para o experimento com PCA. Analisando a FIGURA, é possível observar que a componente 3 foi a que obteve maior valor de importância média, seguido pela componente 2. A componente 1, que mais traduz a variância da base após a transformação do PCA, não apareceu entre as seis com maior importância. Para o experimento com t-SNE, mostrado na FIGURA 123, a componente 2 foi a que mais explicou, na média, o

FIGURA 120 – IMPACTOS DOS HIPERPARÂMETROS NA OTIMIZAÇÃO DO MODELO LR



(a) importância relativa no desempenho



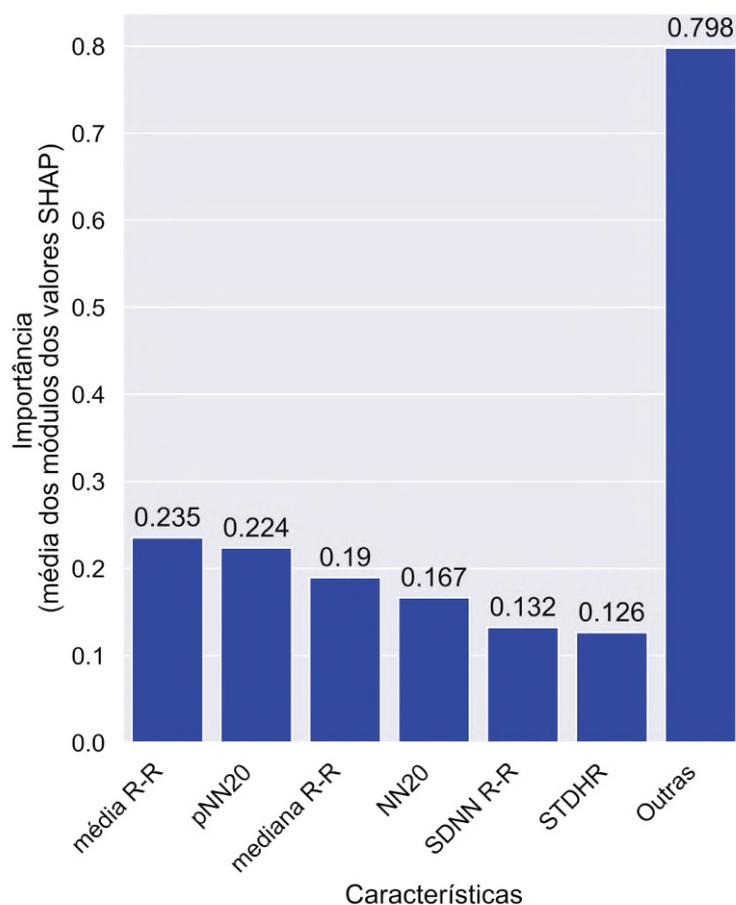
(b) impacto relativo no tempo de otimização

Experimentos  
 sem redução de dimensionalidade  
 PCA  
 t-SNE  
 UMAP

FONTE: Próprio autor  
 LEGENDA: Impactos dos hiperparâmetros no desempenho e no tempo de otimização para modelo LR.

resultado obtido. No experimento com UMAP, a componente 3 foi a que mais contribuiu para explicabilidade do resultado, conforme visto na FIGURA 124.

FIGURA 121 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO LR NO EXPERIMENTO I

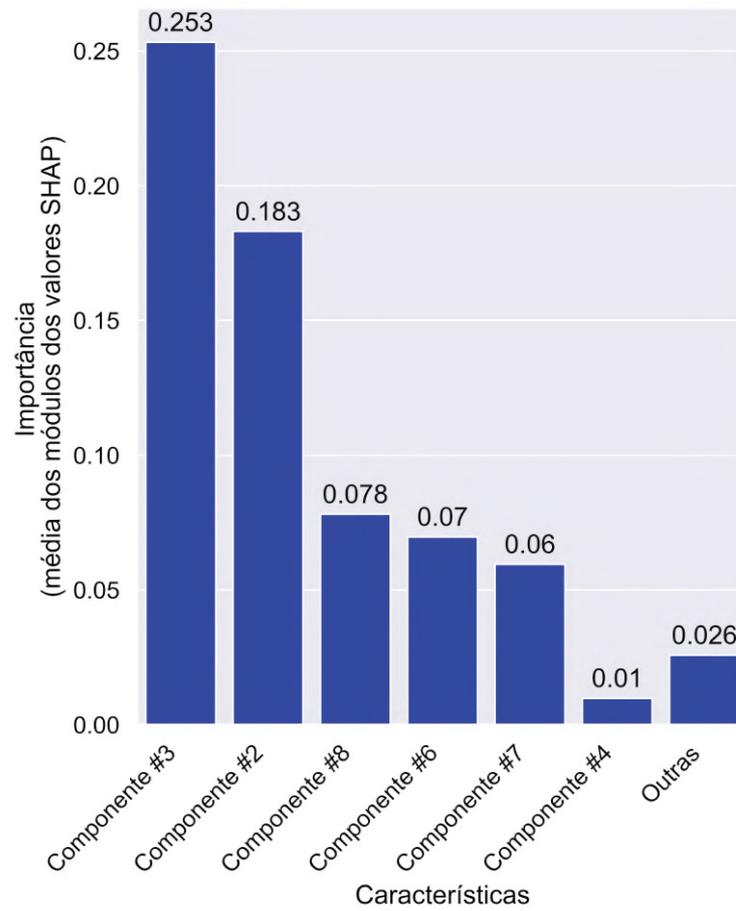


FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo LR no experimento I.

Na análise de como foi o processo de predição, a FIGURA 125 apresenta esta análise para o experimento I. Na parte inferior da FIGURA é possível analisar que as características LF/HF, SDSD R-R, LF, LFnu e HFnu pouco impactaram no processo de decisão. Essa influência aumentou para as características no meio da FIGURA, CVSD, VLF, amplitude R-R, MAXHR, pNN50 e NN50, mas ainda de forma singela se comparada com as características do topo da FIGURA. Estas características, média HR, SDNN R-R, STDHR, NN20, mediana R-R, pNN20 e média R-R, são as que mais impactaram no resultado, com uma influência muito significativa. No experimento II, mostrado na FIGURA 126, é possível observar que as componentes que realmente impactam na decisão do modelo são as componentes sete, seis, oito, dois e três. Com as outras componentes pouco afetando o resultado, salvo algumas exceções, que, geralmente, conduziram para erros de classificação. Para os experimentos III e IV, mostrados nas FIGURAS 127 e 128, respectivamente, o comportamento das linhas de predição é mais homogêneo, com uma componente dominando em cada experimento.

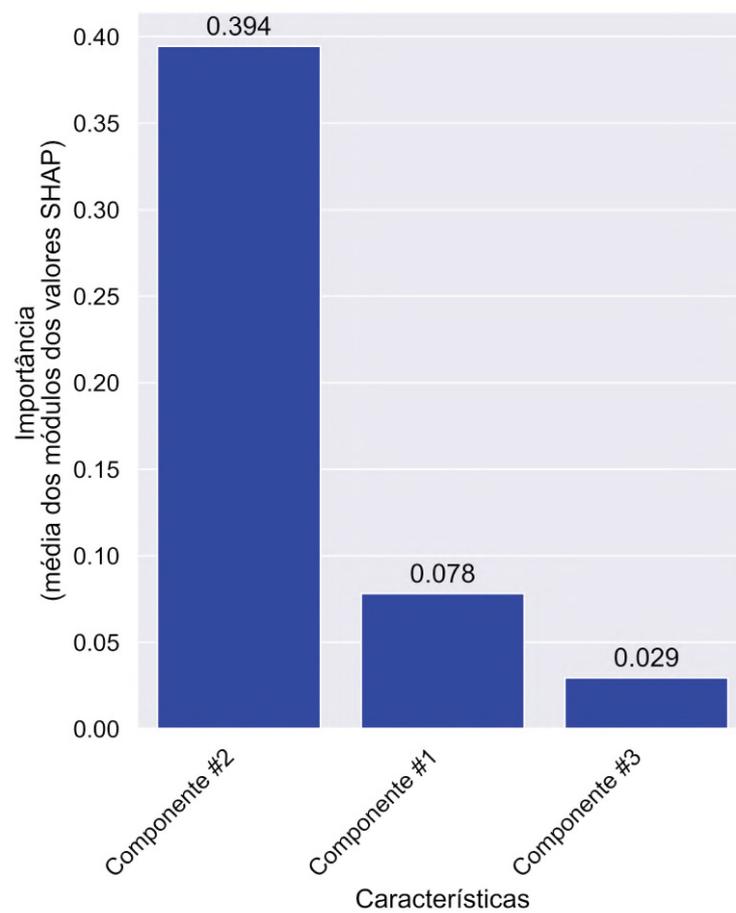
FIGURA 122 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO LR NO EXPERIMENTO II



FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo LR no experimento II.

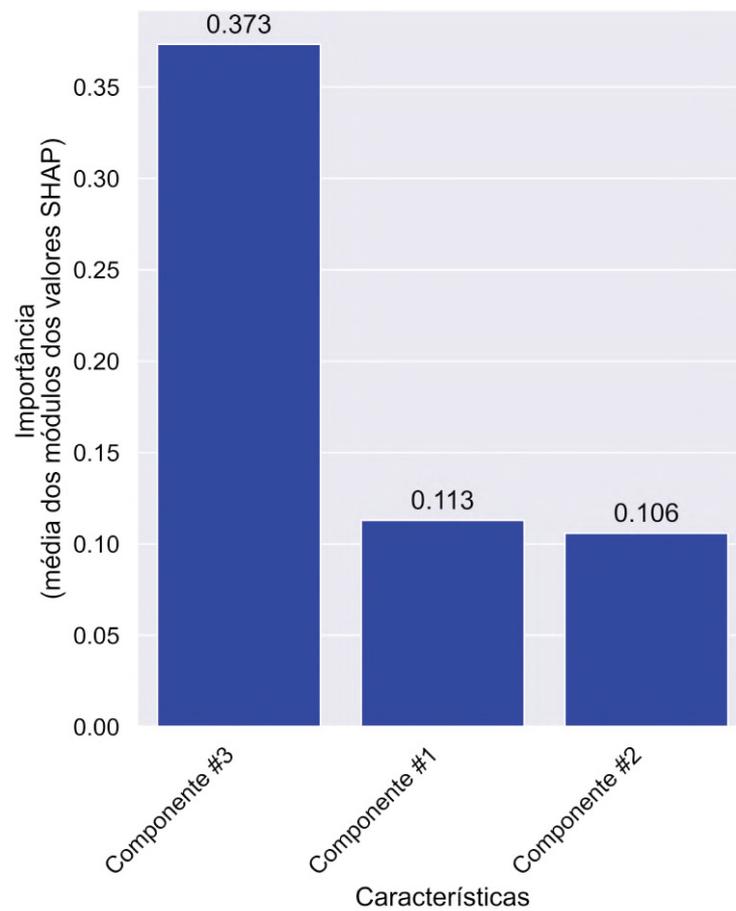
FIGURA 123 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO LR NO EXPERIMENTO III



FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo LR no experimento III.

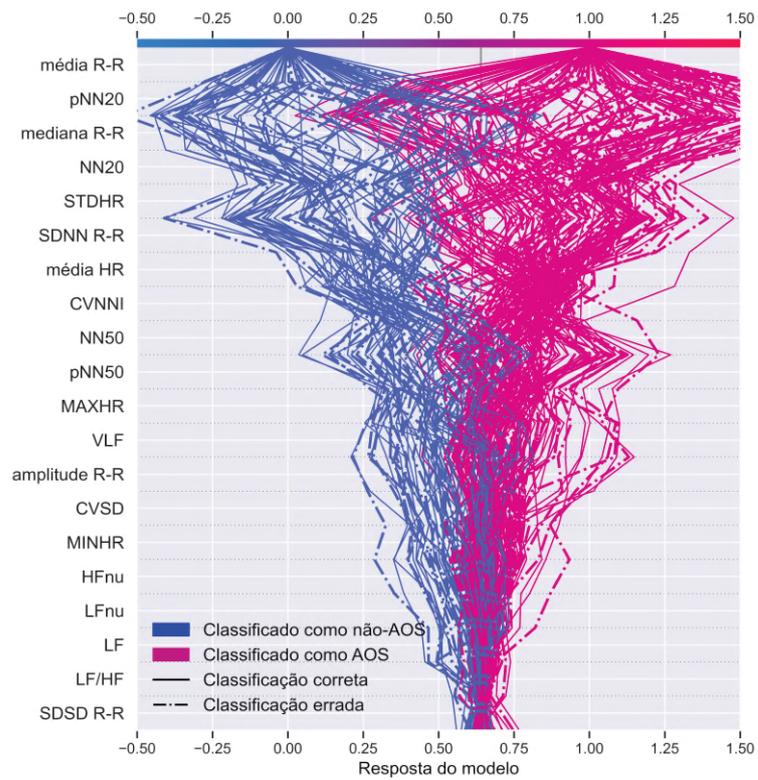
FIGURA 124 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO LR NO EXPERIMENTO IV



FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo LR no experimento IV.

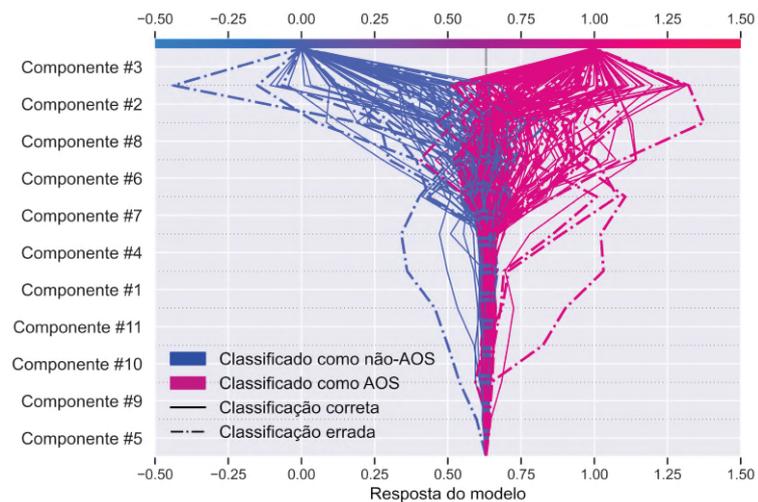
FIGURA 125 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO LR NO EXPERIMENTO I



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo LR no experimento I.

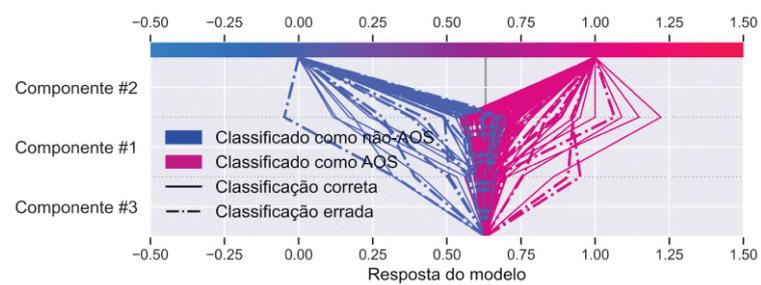
FIGURA 126 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO LR NO EXPERIMENTO II



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo LR no experimento II.

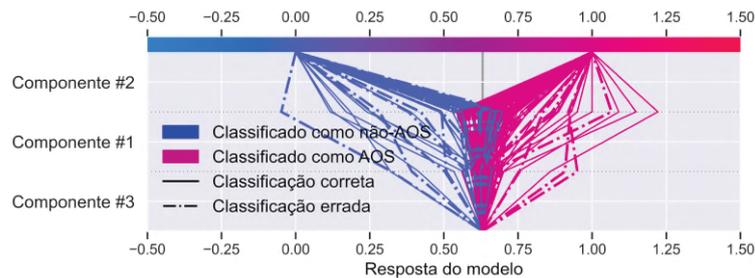
FIGURA 127 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO LR NO EXPERIMENTO III



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo LR no experimento III.

FIGURA 128 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO LR NO EXPERIMENTO IV

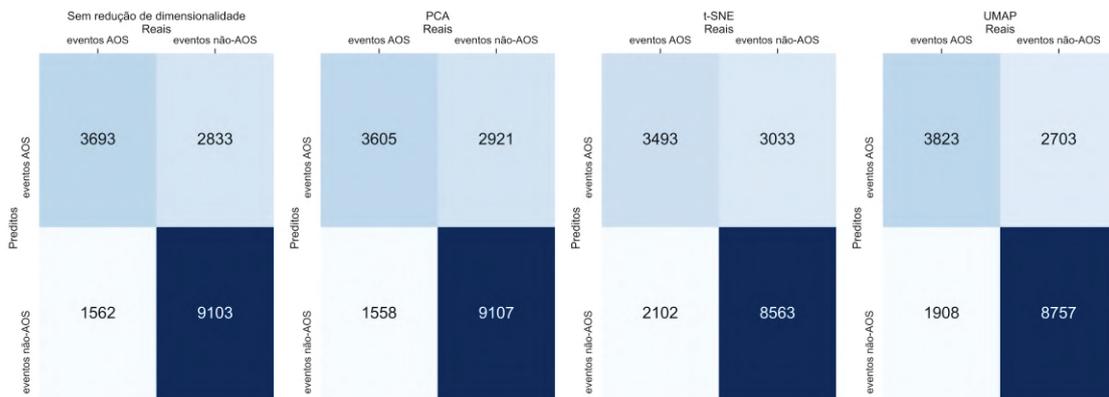


FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo LR no experimento IV.

A FIGURA 129 apresenta as matrizes de confusão do modelo LR para todos os experimentos. Analisando a FIGURA, é possível observar que os modelos de LR, em todos os experimentos, não obtiveram um desempenho satisfatório na classificação de eventos de AOS, sendo possível notar uma grande quantidade de erros do tipo falsos-positivos. O experimento com uso do PCA foi que mais corretamente classificou eventos de não-AOS.

FIGURA 129 – MATRIZ DE CONFUSÃO DO MODELO LR



FONTE: Próprio autor

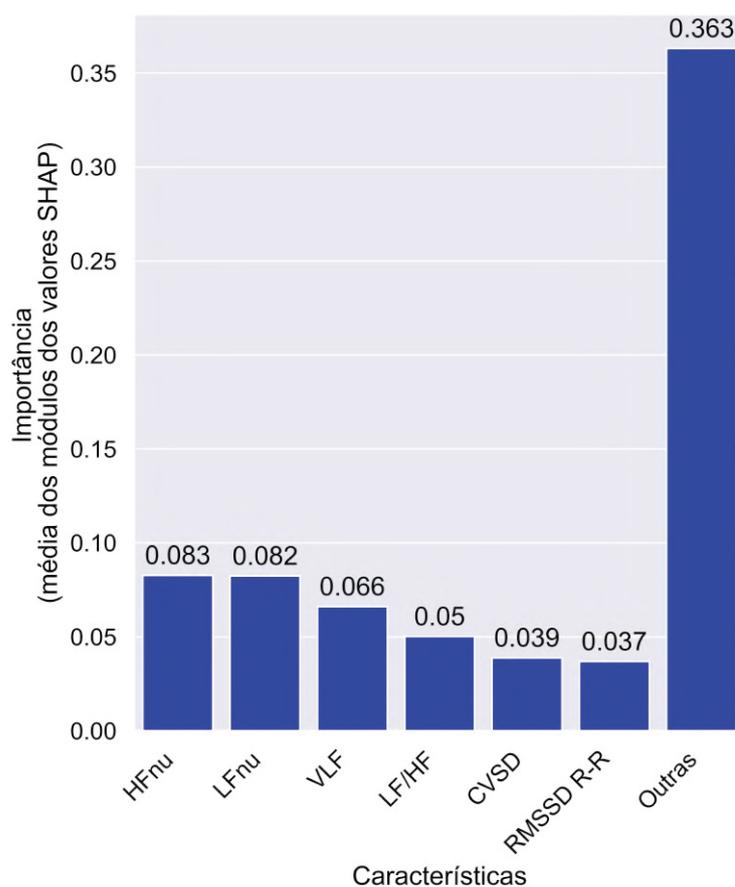
LEGENDA: Matriz de confusão para o modelo LR em cada um dos experimentos.

DEMAIS RESULTADOS DO MODELO NAIVE-BAYES, NB

Como o modelo NB não possui nenhum hiperparâmetro a ser sintonizado, não foram realizados estudos com a ferramenta Optuna. Em termos de explicabilidade, a FIGURA 130 apresenta a importância média das características no experimento I. Na FIGURA, é possível observar que houve um equilíbrio entre as 23 características, com as características HFnu e LFnu sendo as que mais contribuíram para o resultado,

porém não de uma forma dominante. No experimento II, mostrado na imagem 131, as componentes 2 e 3 do PCA tiveram uma contribuição maior e mais significativa que as demais, seguidas pelas componentes 10 e 1. Para o experimento III, representado pelo gráfico da FIGURA 132, a componente 2 foi a mais importante, na média, para explicar o resultado obtido. Com valores semelhantes, a componente 3 do experimento IV foi a que mais explicou o resultado, na média, conforme visto na imagem 133.

FIGURA 130 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO NB NO EXPERIMENTO I

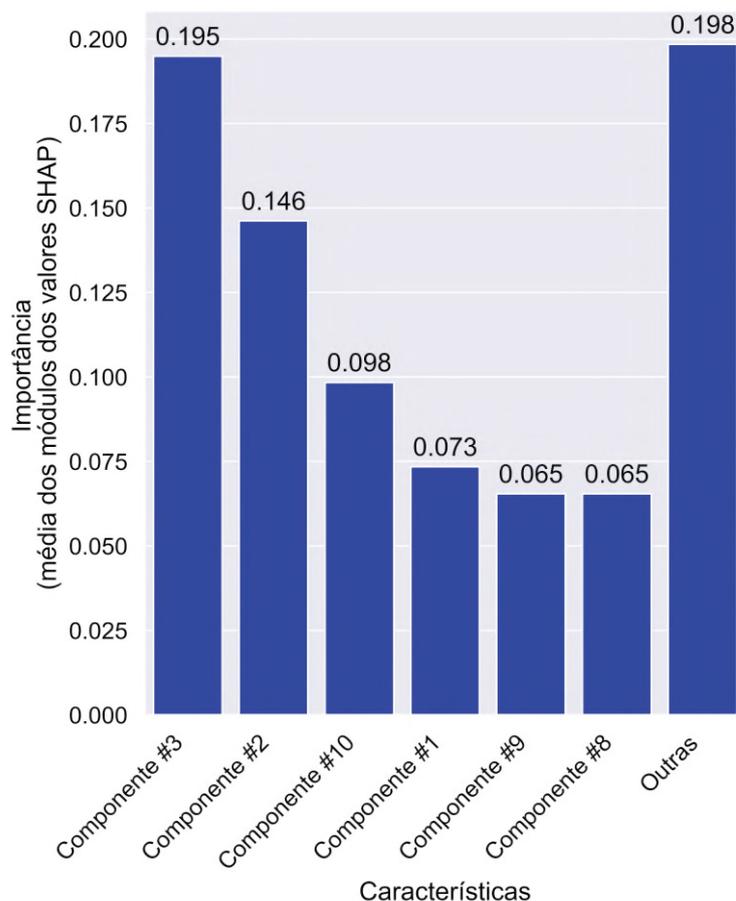


FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo NB no experimento I.

No processo de inferência do modelo, a FIGURA 134 apresenta para o experimento sem redução de dimensionalidade. Na FIGURA, é possível observar que para classificação dos eventos de não-AOS, as características da parte inferior e da parte central da FIGURA vão adicionando pequenos incrementos em direção ao resultado modelo, com as características do topo da FIGURA responsáveis pelos maiores incrementos. Para eventos de AOS, as características da parte central da FIGURA tem maiores influências, adicionando mais à resposta do modelo. As características do topo continuam fazendo as maiores adições. Também é possível observar que o valor inicial

FIGURA 131 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO NB NO EXPERIMENTO II



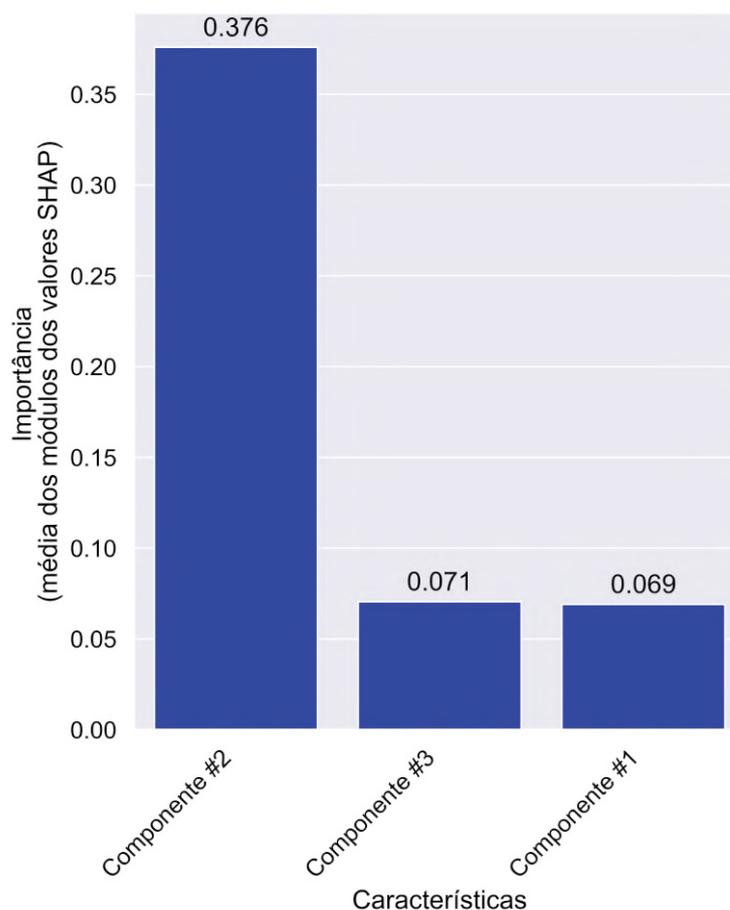
FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo NB no experimento II.

da resposta do modelo é aproximadamente 0,30, de modo que, inicialmente, a resposta do modelo tende a um evento de não-AOS. Este comportamento não foi observado no experimento com PCA, da FIGURA 135, em que o valor inicial ficou próximo de 0,70. Neste experimento, as componentes 3, 2 e 10, do topo da FIGURA, foram as que mais influenciaram a decisão do modelo. No experimento com t-SNE, mostrado na FIGURA 136, é possível observar que as três componentes tem um papel significativo para predição de eventos de AOS, enquanto que para predição de eventos de não-AOS, a componente 2 apresenta maior influência, na média. Um comportamento mais homogêneo é observado no experimento com aplicação do UMAP, da FIGURA 137, com a componente 3 tendo o maior impacto na resposta do modelo para predição de ambas as classes.

O efeito da resposta do modelo ter uma espécie de viés ao redor dos eventos de não-AOS é visto na matriz de confusão do experimento I, mostrada na FIGURA 138, juntamente com as matrizes de confusão dos demais experimentos. No experimento I é

FIGURA 132 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO NB NO EXPERIMENTO III



FONTE: Próprio autor

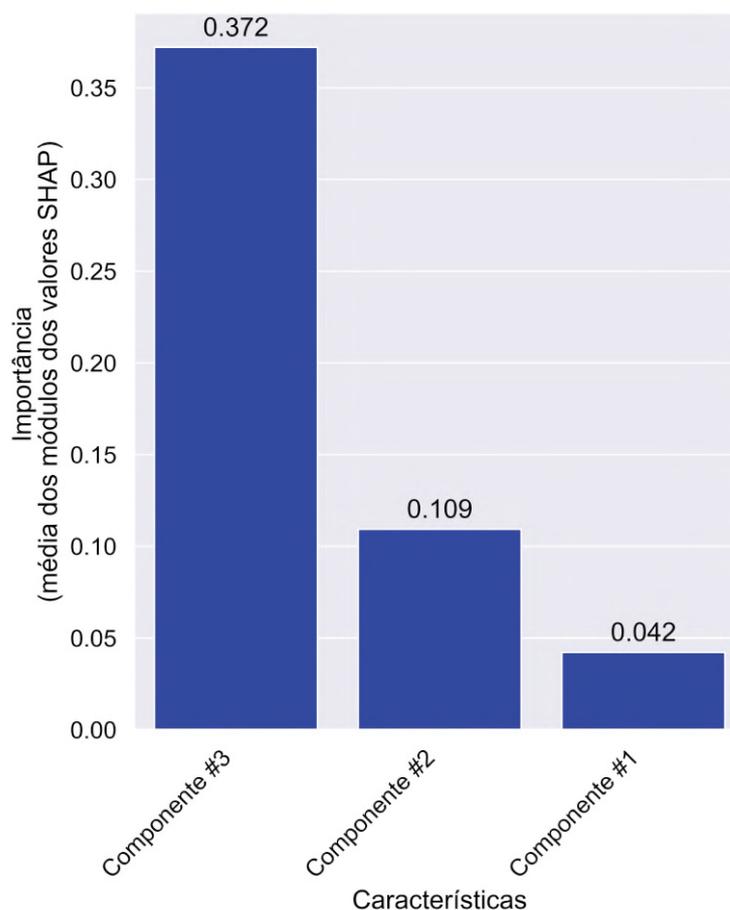
LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo NB no experimento III.

possível observar que o modelo teve um viés maior para prever eventos de não-AOS. No entanto, este viés se deu de uma forma incorreta, de modo que este experimento, juntamente com experimento do UMAP, foi o que mais classificou eventos como não-AOS e o que mais errou nesta classificação. O experimento com UMAP, foi o que mais acertou estes eventos. Para eventos de AOS, o experimento sem redução de dimensionalidade foi o que mais classificou corretamente estes eventos.

#### DEMAIS RESULTADOS DO MODELO FLORESTA ALEATÓRIA, RF

O gráfico da imagem 139 apresenta o histórico de otimização no experimento sem redução de dimensionalidade para o modelo RF. Na FIGURA, é possível observar que a maioria dos resultados ficaram entre 0,82 e 0,84, com algumas configurações obtendo um resultado abaixo de 0,82 e acima de 0,78. A FIGURA 140 apresenta o mesmo gráfico para experimento com PCA, o comportamento foi, de forma geral, semelhante ao do experimento anterior, com a diferença que o intervalo de variação das melhores

FIGURA 133 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO NB NO EXPERIMENTO IV



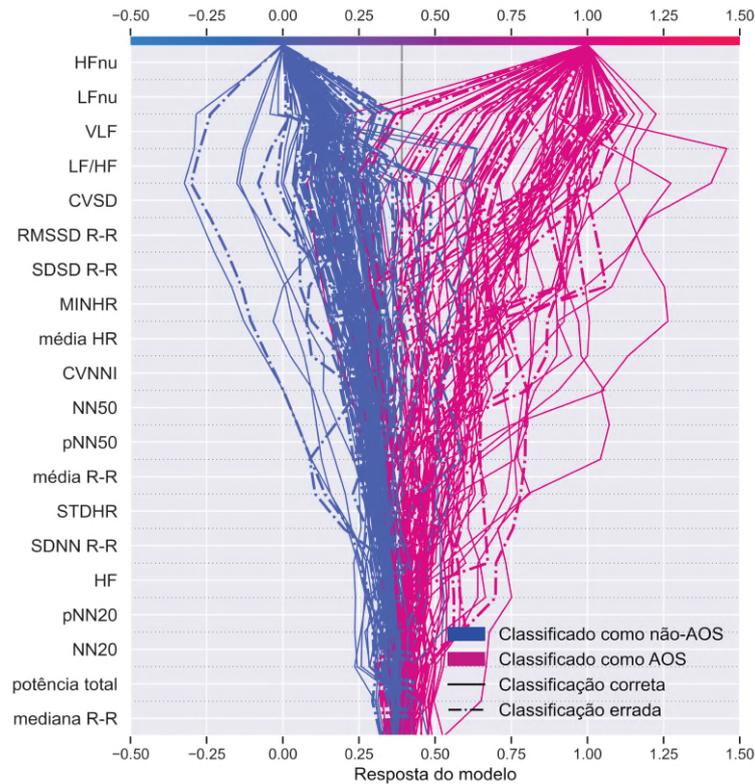
FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo NB no experimento IV.

configurações foi menor, entre 0,83 e 0,84. Similarmente, a FIGURA 141 apresenta o mesmo comportamento mas com uma variação menor, entre 0,80 e 0,81 para as melhores configurações, no experimento do t-SNE. Finalmente, na FIGURA 142, é possível observar uma variação maior, tanto dos melhores resultados, quanto dos piores. Ao todo, com os resultados entre 0,77 e 0,80.

A imagem 143 mostra, no gráfico "a" (em cima), a importância relativa de cada hiperparâmetro, enquanto o gráfico indicado por "b" (em baixo) mostra o impacto relativo no tempo de otimização para cada um dos experimentos. Em todos os experimentos, o hiperparâmetro profundidade máxima desempenhou a maior importância, enquanto o hiperparâmetro número de estimadores foi o que mais impactou o tempo, mas com um efeito muito pequeno no resultado do treinamento. Isso pode levar a novos estudos, em que mais configurações da profundidade máxima são testadas enquanto o valor do número de estimadores é deixado constante ou com um espaço de busca reduzido.

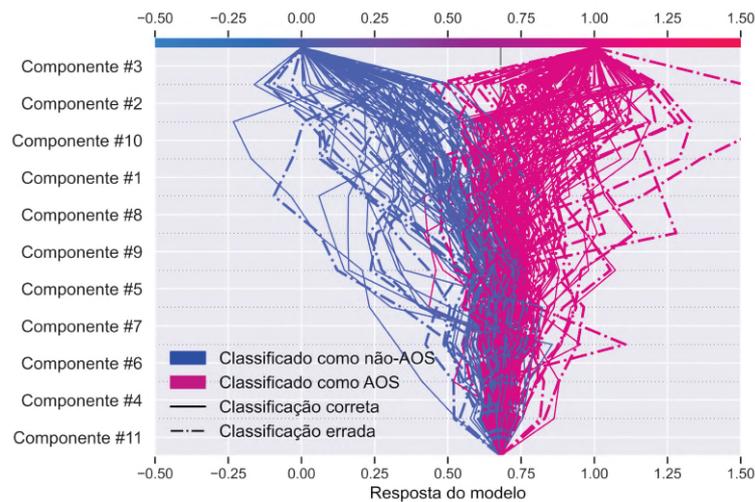
FIGURA 134 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO NB NO EXPERIMENTO I



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo NB no experimento I.

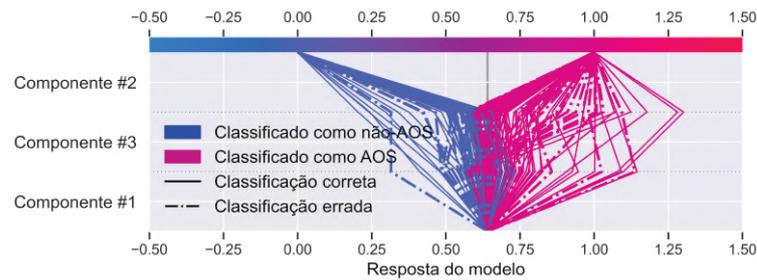
FIGURA 135 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO NB NO EXPERIMENTO II



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo NB no experimento II.

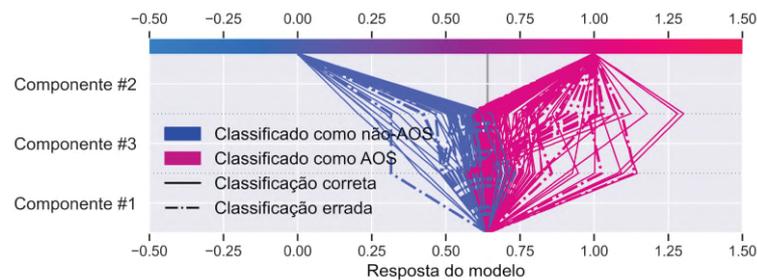
FIGURA 136 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO NB NO EXPERIMENTO III



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo NB no experimento III.

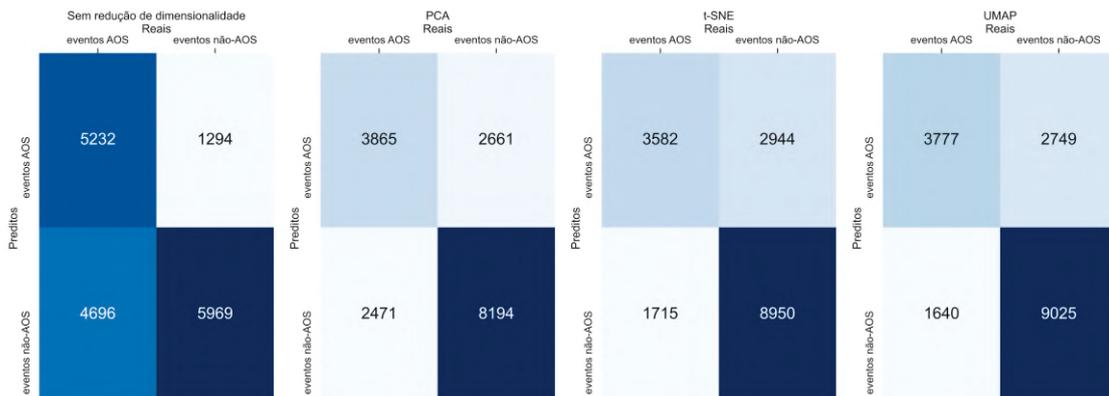
FIGURA 137 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO NB NO EXPERIMENTO IV



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo NB no experimento IV.

FIGURA 138 – MATRIZ DE CONFUSÃO DO MODELO NB

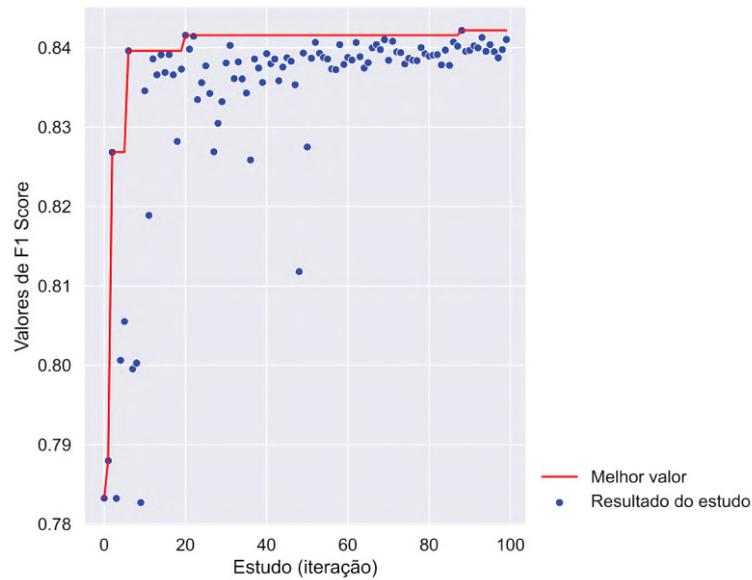


FONTE: Próprio autor

LEGENDA: Matriz de confusão para o modelo NB em cada um dos experimentos.

Em termos de explicabilidade, a característica VLF foi a que mais contribuiu e explicou o resultado do modelo RF na base de teste no experimento sem redução de dimensionalidade, com uma média de módulos dos valores SHAP em cerca de três vezes o valor da característica pNN20, segunda colocada, conforme mostra a imagem da FIGURA 144. Enquanto que neste experimento uma característica teve

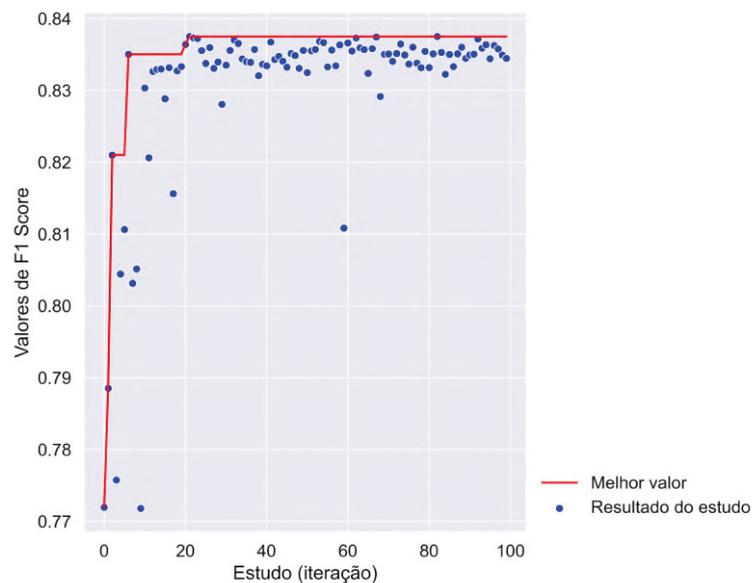
FIGURA 139 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO RF NO EXPERIMENTO I



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de RF no experimento I.

FIGURA 140 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO RF NO EXPERIMENTO II

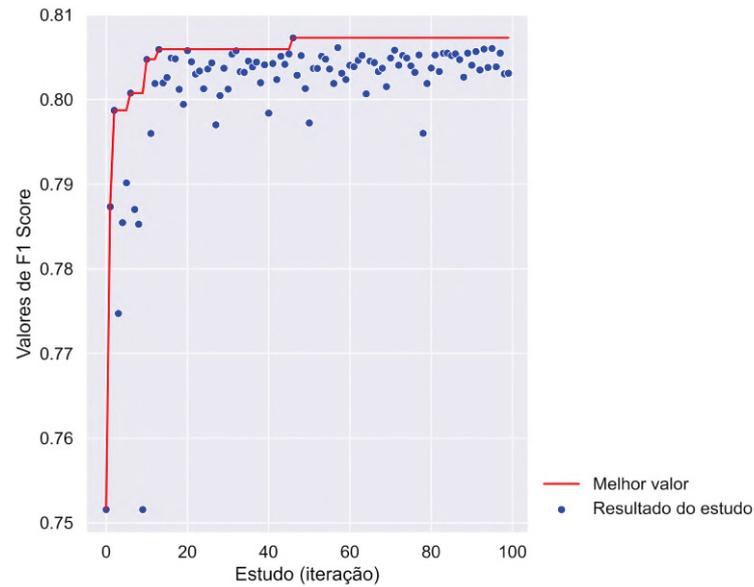


FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de RF no experimento II.

uma importância maior, no experimento com aplicação do PCA, mostrado na FIGURA 145, as componentes 3 e 2 foram as que mais contribuíram para o resultado, seguidas das componentes 8 e 1. No experimento com uso do t-SNE da FIGURA 146, houve um equilíbrio maior entre as componentes, uma vez que neste experimento a base de dados contava com apenas três dimensões. A componente 2 foi a que obteve maior

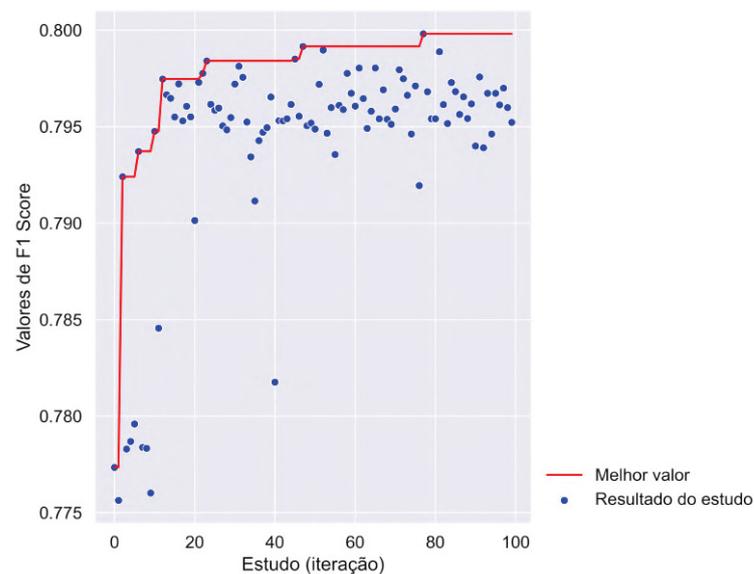
FIGURA 141 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO RF NO EXPERIMENTO III



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de LR no experimento III.

FIGURA 142 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO RF NO EXPERIMENTO IV



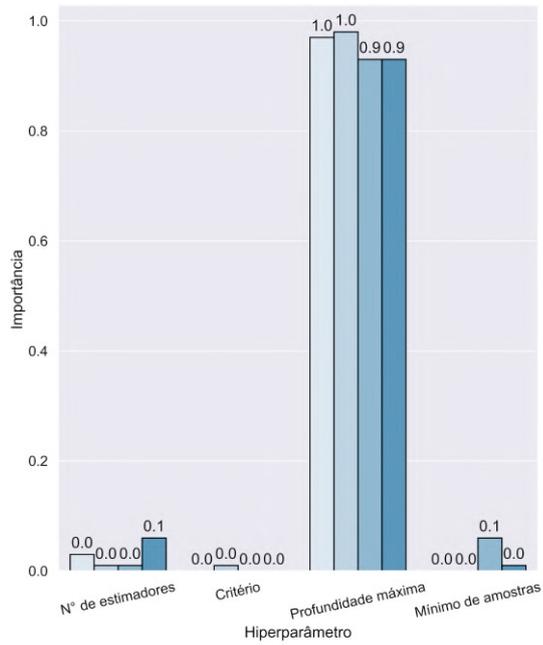
FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de RF no experimento IV.

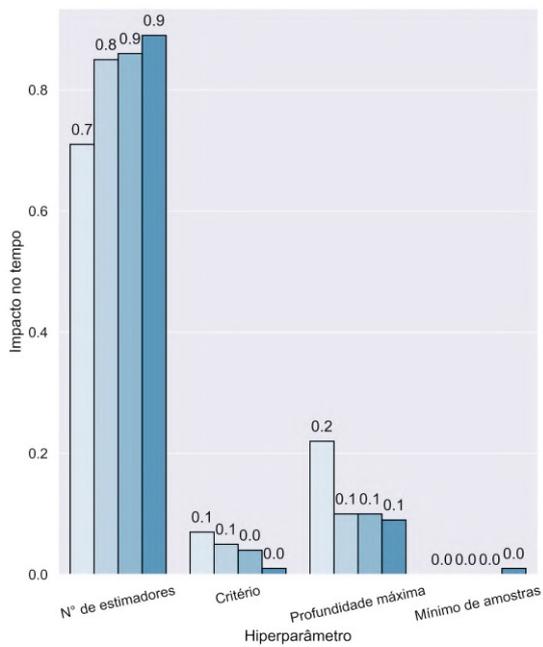
importância. Este mesmo comportamento é observado de forma similar na FIGURA 147, referente ao experimento com UMAP.

Para o processo de decisão, o gráfico da FIGURA 148 mostra que as características da parte inferior da FIGURA pouco contribuíram para a média dos resultados, enquanto qu as do topo, especialmente as três do topo, contribuíram de forma mais

FIGURA 143 – IMPACTOS DOS HIPERPARÂMETROS NA OTIMIZAÇÃO DO MODELO RF



(a) importância relativa no desempenho



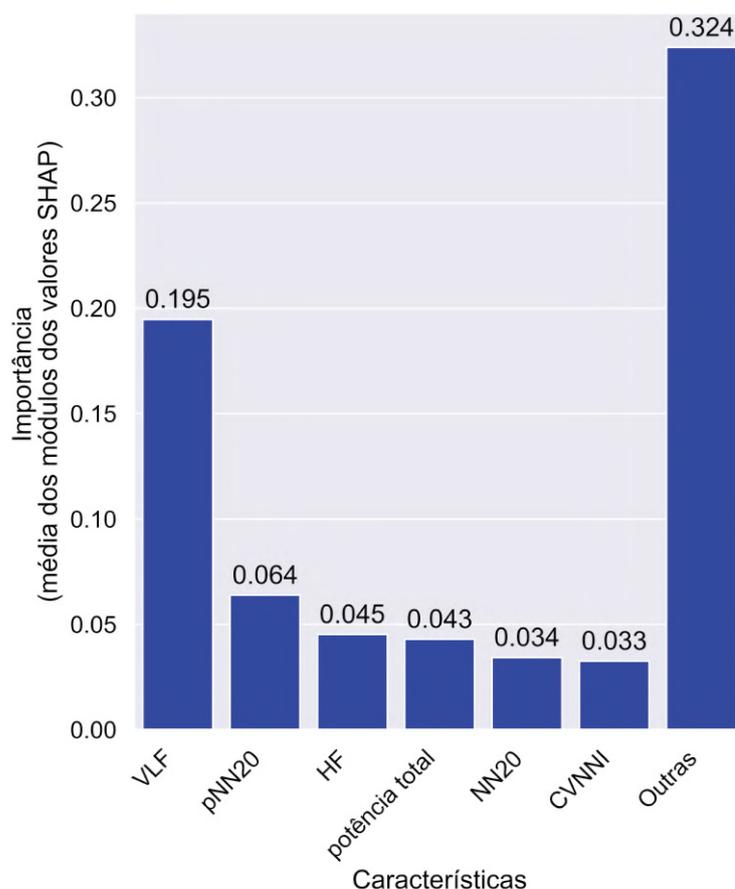
(b) impacto relativo no tempo de otimização

- Experimentos
- sem redução de dimensionalidade
  - PCA
  - t-SNE
  - UMAP

FONTE: Próprio autor  
 LEGENDA: Impactos dos hiperparâmetros no desempenho e no tempo de otimização para modelo RF.

significativa para o resultado. Uma observação importante é em relação a algumas predições erradas em que valores elevados das características da parte inferior da

FIGURA 144 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO RF NO EXPERIMENTO I



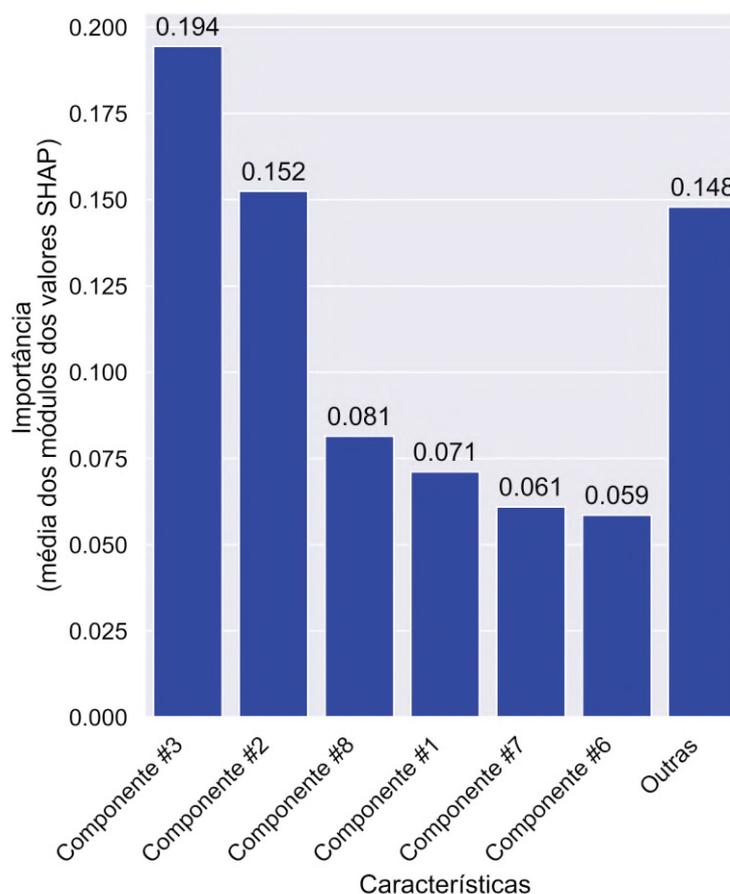
FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo RF no experimento I.

FIGURA tiveram um papel significativo no erro do modelo. Um comportamento similar é observado no gráfico da imagem 149, onde este processo de inferência é representado para experimento com PCA, da parte inferior da FIGURA até a componente 8, as componentes, na média, pouco contribuíram para a decisão do modelo, coma exceção de algumas predições erradas, em que valores elevados destas características também fizeram com o modelo errasse ao distinguir as classes. Devido a um número reduzido de dimensões, tanto para o experimento aplicando t-SNE, mostrado na FIGURA 150, quanto o experimento com uso de UMAP, mostrado na FIGURA 151, o comportamento das linhas de predições é mais homogêneo.

A imagem 152 apresenta as matrizes de confusão do modelo de RF. O maior número de eventos de AOS classificados corretamente foi no experimento I, seguido pelos experimento II, III e IV respectivamente. O maior número de eventos de não-AOS, no entanto, foi com experimento II. Uma hipótese a ser analisada neste caso é se combinar um classificador sem redução de dimensionalidade testando eventos de AOS

FIGURA 145 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO RF NO EXPERIMENTO II



FONTE: Próprio autor

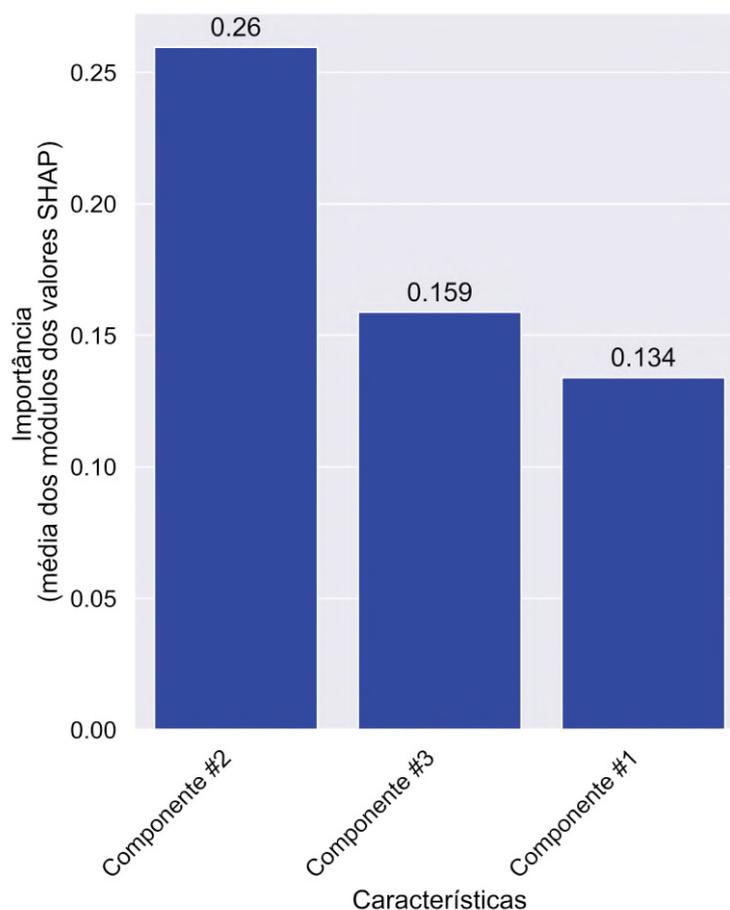
LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo RF no experimento II.

e um classificador com PCA testando eventos de não-AOS, aumentaria a assertividade como um todo.

#### DEMAIS RESULTADOS DO MODELO *EXTREME GRADIENT BOOSTING*, XGBOOST

De modo geral, o histórico de otimização do modelo XgBoost teve uma grande variação em todos os experimentos. Na FIGURA 153 que mostra o histórico para o experimento I, é possível observar que muitas configurações se aproximaram do melhor valor, mas que outras configurações apresentaram um resultado que varou de 0,80 até cerca de 0,65, com alguns resultados podendo ser caracterizados como *outliers*. Esta grande variância também é observada na FIGURA 154, do experimento II, em que os resultados foram de 0,80 até cerca de 0,50, com uma configuração em que o modelo resultou em um subajuste ou *underfitting*. Algo similar também ocorreu no experimento III, mostrado no gráfico da imagem 155, em que dois casos de *underfitting* são observados, com resultados iguais a cerca de 0 e 0,40. Na FIGURA

FIGURA 146 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO RF NO EXPERIMENTO III



FONTE: Próprio autor

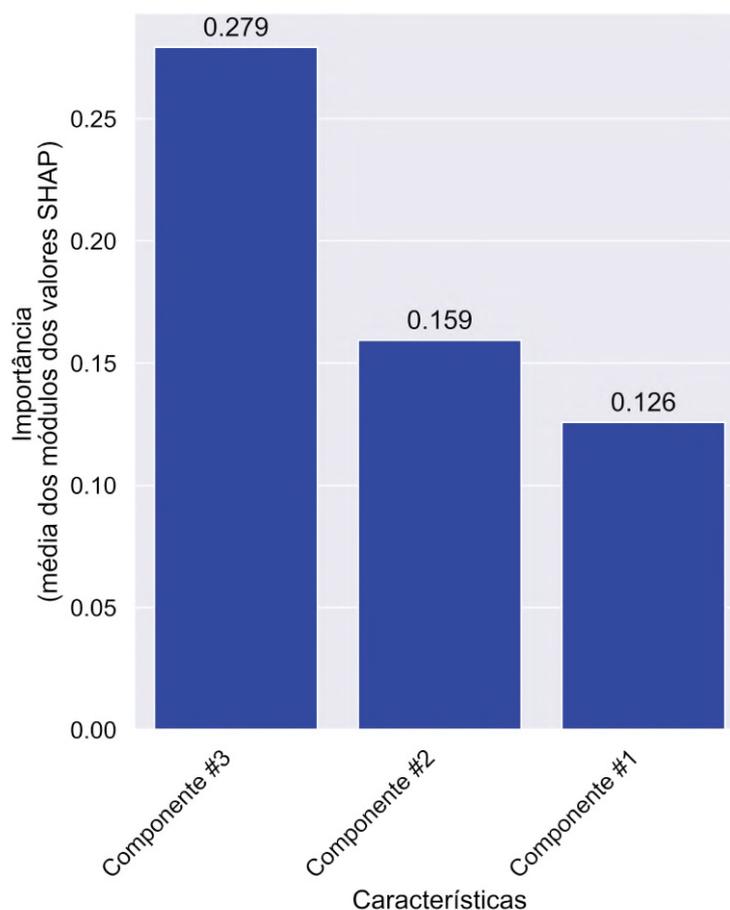
LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo RF no experimento III.

156, que apresenta o gráfico para o experimento IV, embora um resultado tenha desempenhado um F1-score de aproximadamente 0,55, não houve resultados abaixo de 0,50.

A FIGURA 157 apresenta o impacto dos hiperparâmetros do XgBoost. O impacto relativo do resultado, gráfico à direita da FIGURA, a taxa de aprendizado foi o hiperparâmetro que mais teve importância no resultado. A razão de amostragem também teve um significativo impacto no experimento com PCA, enquanto que a profundidade máxima afetou o experimento com PCA e com t-SNE. No quesito impacto no tempo de otimização, com exceção do experimento com t-SNE, a profundidade máxima foi a que mais teve impacto no tempo. Para o experimento com t-SNE, o número de estimadores foi o que mais impactou o tempo.

A FIGURA 158 mostra a média de importância de cada uma das características no experimento I. Analisando a FIGURA, embora nenhuma característica tenha

FIGURA 147 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO RF NO EXPERIMENTO IV



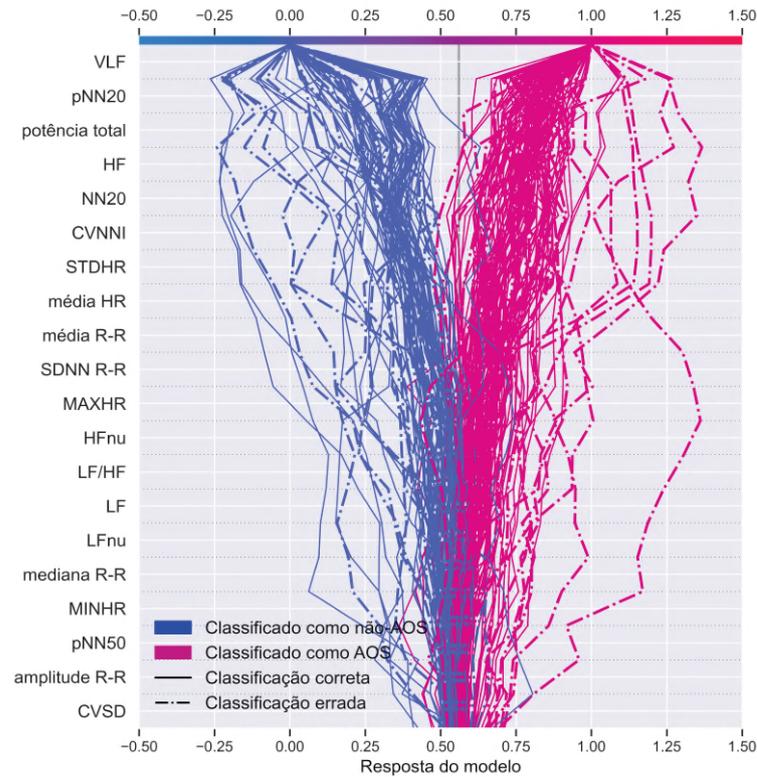
FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo RF no experimento IV.

dominado o resultado, a característica VLF foi a que mais contribuiu para o resultado do modelo, seguido pelas características pNN20 e mediana R-R. Este certo equilíbrio entre a importância média das características pode ter acontecido devido ao fato da XgBoost construir diferentes estimadores que vão analisar diferentes conjuntos de características. No experimento II da FIGURA 161, as três primeiras componentes do PCA foram as que mais impactaram o resultado, na ordem da componente 3, seguido da 2 e por fim a 1. A FIGURA 160 apresenta para experimento III, onde a componente 2 foi a que mais teve importância no resultado do modelo, enquanto que na FIGURA 161, do experimento IV, foi a componente 3.

A FIGURA 162 apresenta o processo de inferência para as 200 amostras avaliadas no experimento sem redução de dimensão. Na FIGURA, é possível observar que a separação dos eventos começa a ficar evidente no meio da FIGURA, com as características NN20, STDHR, SDNN R-R, MAXHR e SDSD R-R. Depois dessa separação, praticamente não existe nenhuma linha de predição que acabe mudando de

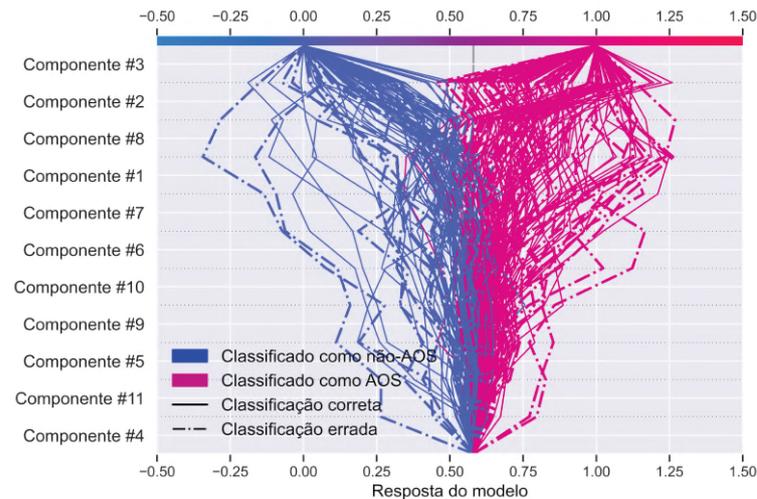
FIGURA 148 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO RF NO EXPERIMENTO I



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo RF no experimento I.

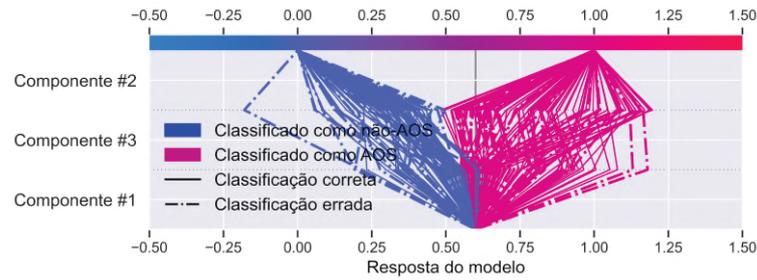
FIGURA 149 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO RF NO EXPERIMENTO II



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo RF no experimento II.

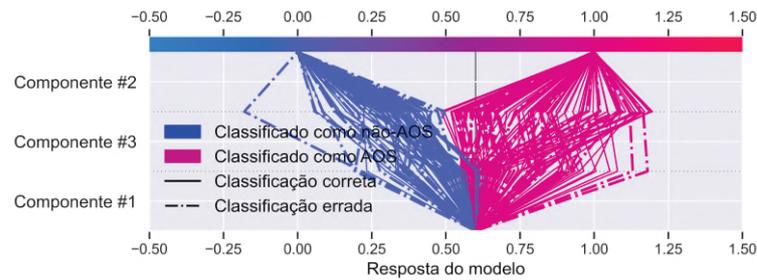
FIGURA 150 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO RF NO EXPERIMENTO III



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo RF no experimento III.

FIGURA 151 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO RF NO EXPERIMENTO IV



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo RF no experimento IV.

FIGURA 152 – MATRIZ DE CONFUSÃO DO MODELO RF

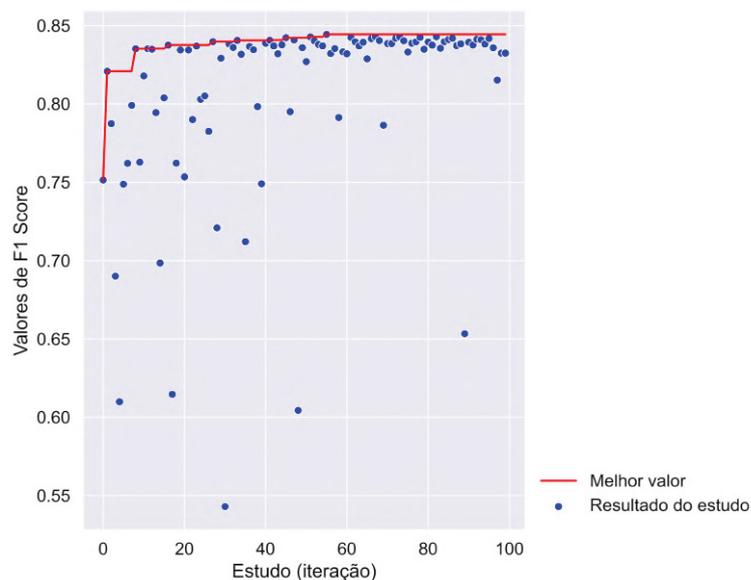


FONTE: Próprio autor

LEGENDA: Matriz de confusão para o modelo RF em cada um dos experimentos.

classe. O mesmo processo para o experimento com PCA é mostrado na 163. Diferente do observado no experimento passado, nesta FIGURA as classes são separadas já no topo da FIGURA, com as componentes 8, 1, 2 e 3. Para o experimento com aplicação do t-SNE da FIGURA 164, é possível notar que as classes são separadas na componente da parte inferior da FIGURA, a componente 1. Poucas predições são

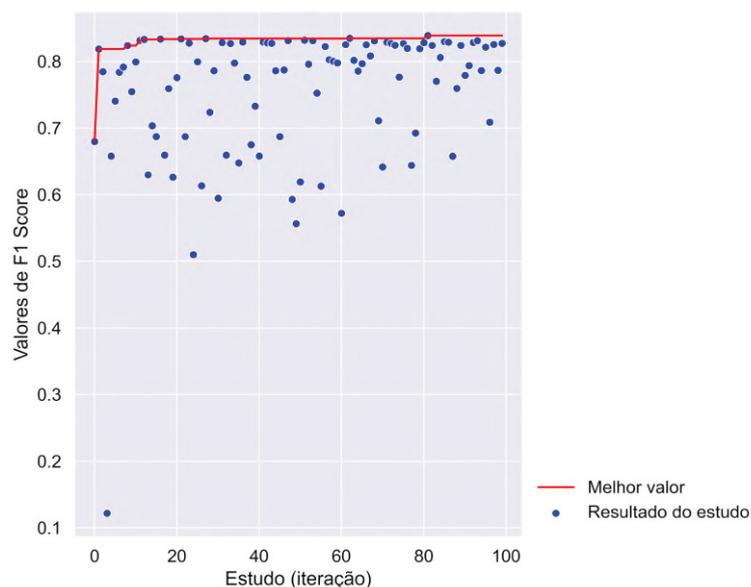
FIGURA 153 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO XGBOOST NO EXPERIMENTO I



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de XgBoost no experimento I.

FIGURA 154 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO XGBOOST NO EXPERIMENTO II



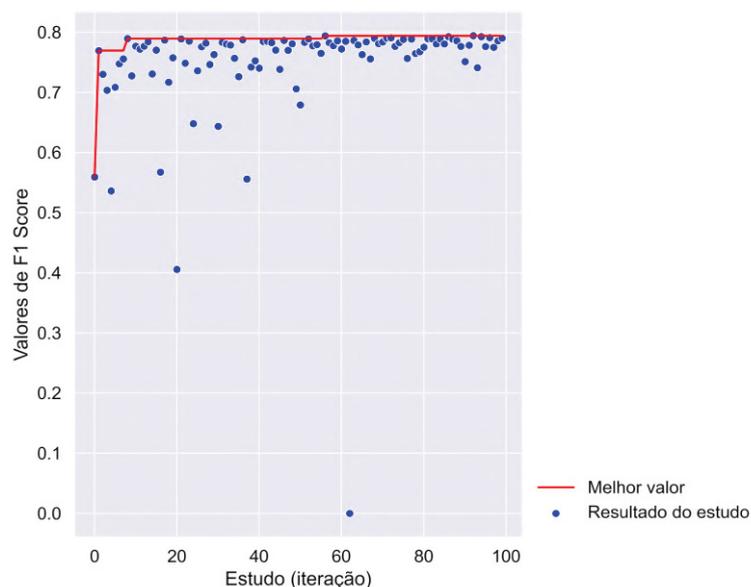
FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de XgBoost no experimento II.

alteradas (terminam com rótulo diferente) após esta avaliação. No experimento com UMAP, FIGURA 165, a separação das classes realmente acontece após a avaliação da componente 3.

A imagem da FIGURA 166 mostra as matrizes de confusão. O experimento

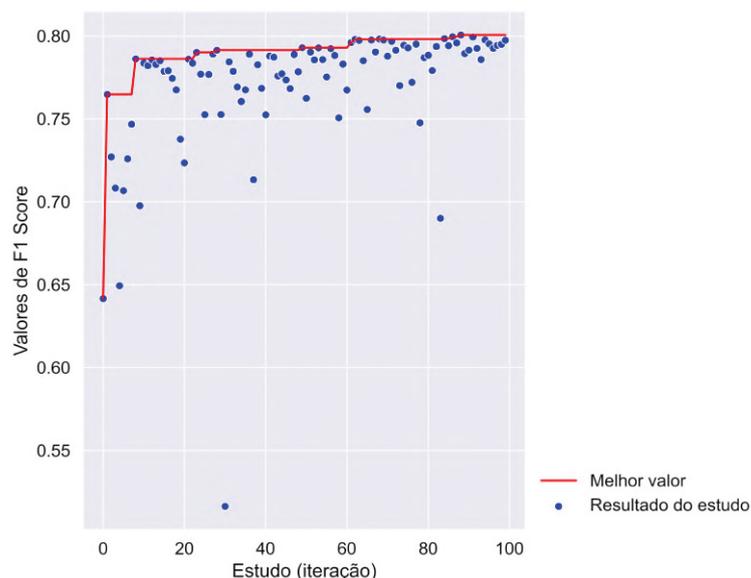
FIGURA 155 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO XGBOOST NO EXPERIMENTO III



FONTE: Próprio autor

LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de XgBoost no experimento III.

FIGURA 156 – HISTÓRICO DE OTIMIZAÇÃO DO MODELO XGBOOST NO EXPERIMENTO IV

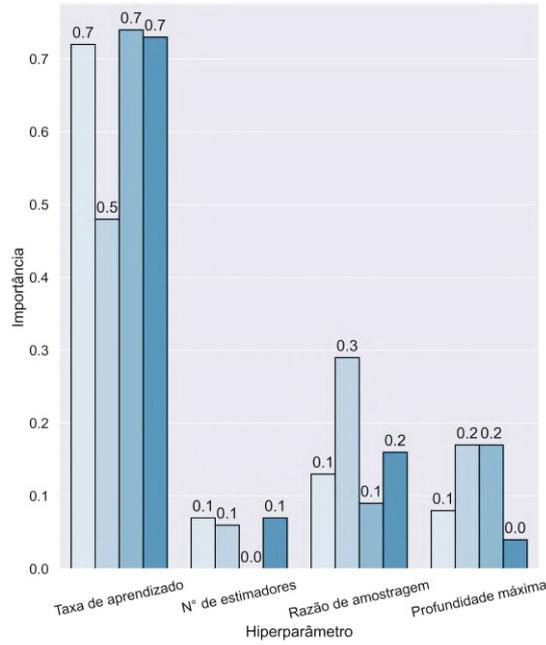


FONTE: Próprio autor

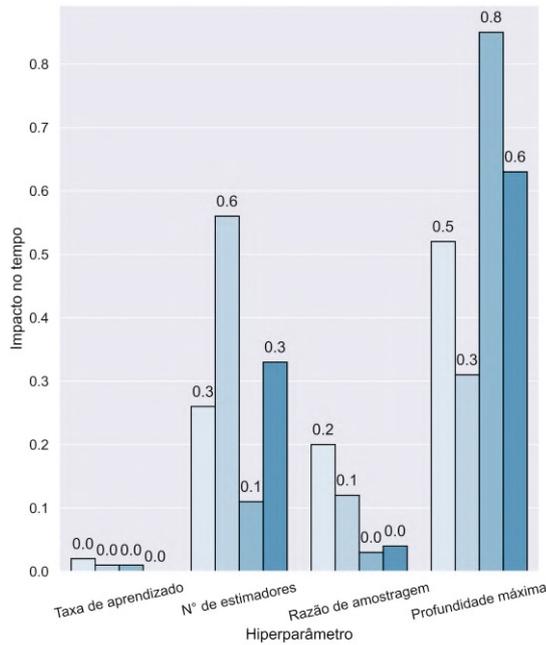
LEGENDA: Histórico de otimização com os resultados obtidos em cada iteração para o modelo de XgBoost no experimento IV.

sem redução de dimensionalidade foi o que melhor classificou corretamente eventos de AOS, enquanto o experimento com t-SNE o que menos classificou estes eventos corretamente. Para os eventos de não-AOS, o experimento com uso de UMAP se destacou com maior número de acertos.

FIGURA 157 – IMPACTOS DOS HIPERPARÂMETROS NA OTIMIZAÇÃO DO MODELO XG-BOOST



(a) importância relativa no desempenho

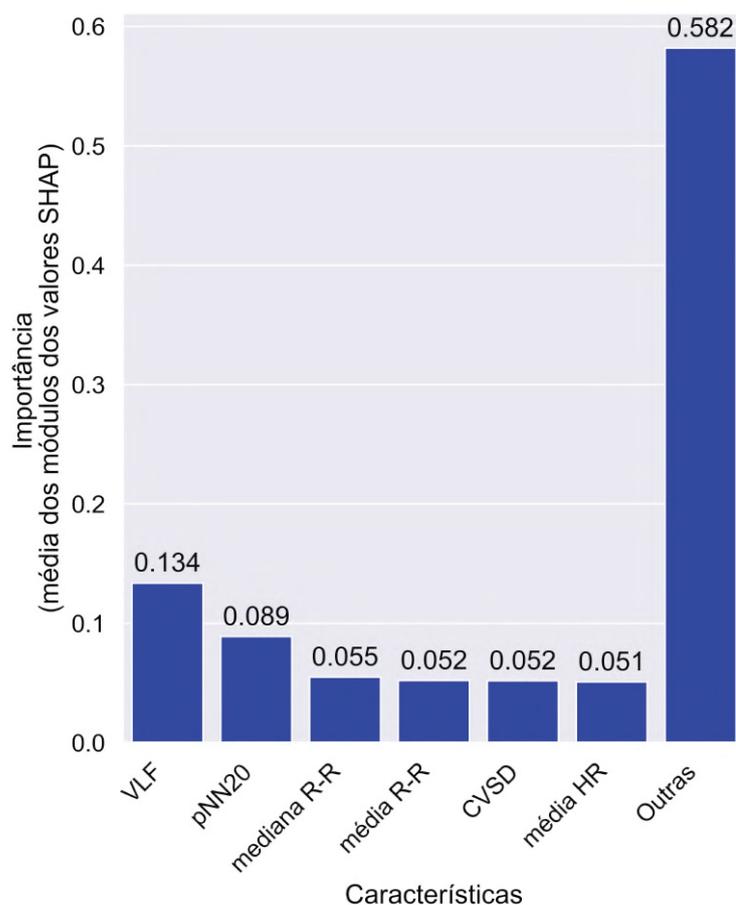


(b) impacto relativo no tempo de otimização

- Experimentos
- sem redução de dimensionalidade
  - PCA
  - t-SNE
  - UMAP

FONTE: Próprio autor  
 LEGENDA: Impactos dos hiperparâmetros no desempenho e no tempo de otimização para modelo XgBoost.

FIGURA 158 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO XGBOOST NO EXPERIMENTO I



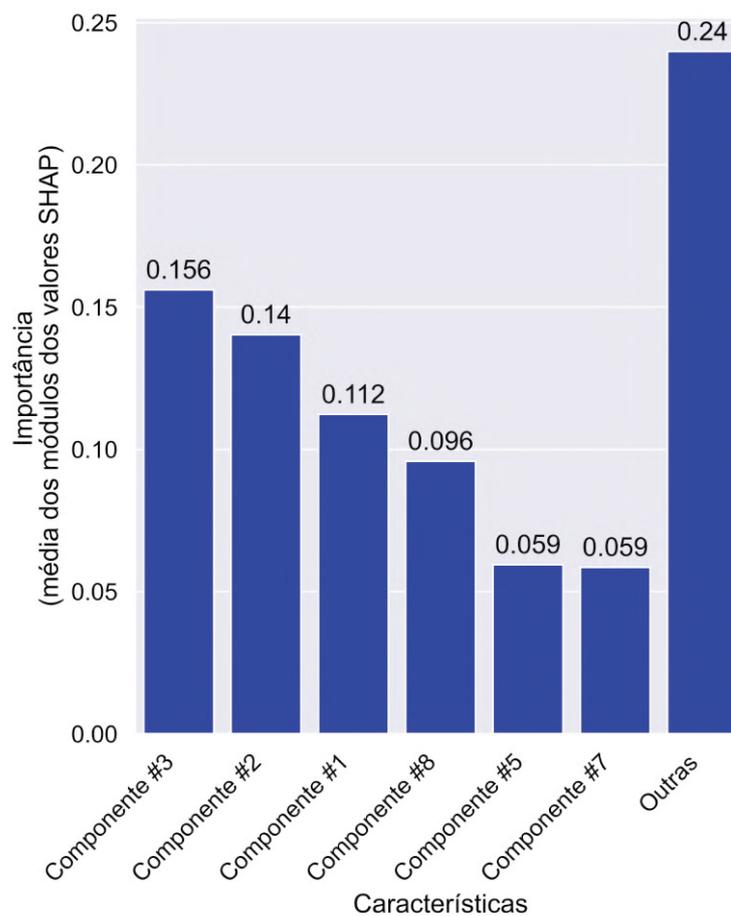
FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo XgBoost no experimento I.

## RESULTADOS DOS DEMAIS TESTES DE HIPÓTESES

As TABELAS 70 e 71 apresentam os resultados dos testes de hipóteses e os tempos totais de predição de toda a base de teste para os modelos de ML. Os modelos foram divididos nas TABELAS, de modo que cada coluna é um dos modelos estudados, enquanto as linhas contém o tempo de predição e os resultados do teste binomial e do teste de McNema, separados para cada experimento. Todos os p-valores encontrados, tanto no teste binomial quanto no teste McNema, estão muito abaixo do alfa estabelecido ( $<0,005$ ), portando é descartado a hipóteses de que os modelos encontraram o resultado obtido por puro acaso. Os tempos de predição foram, em geral, pequenos, com o modelo VM apresentando os maiores tempos. O maior tempo registrado foi de cerca de 130 segundos no experimento com UMAP para o modelo VM.

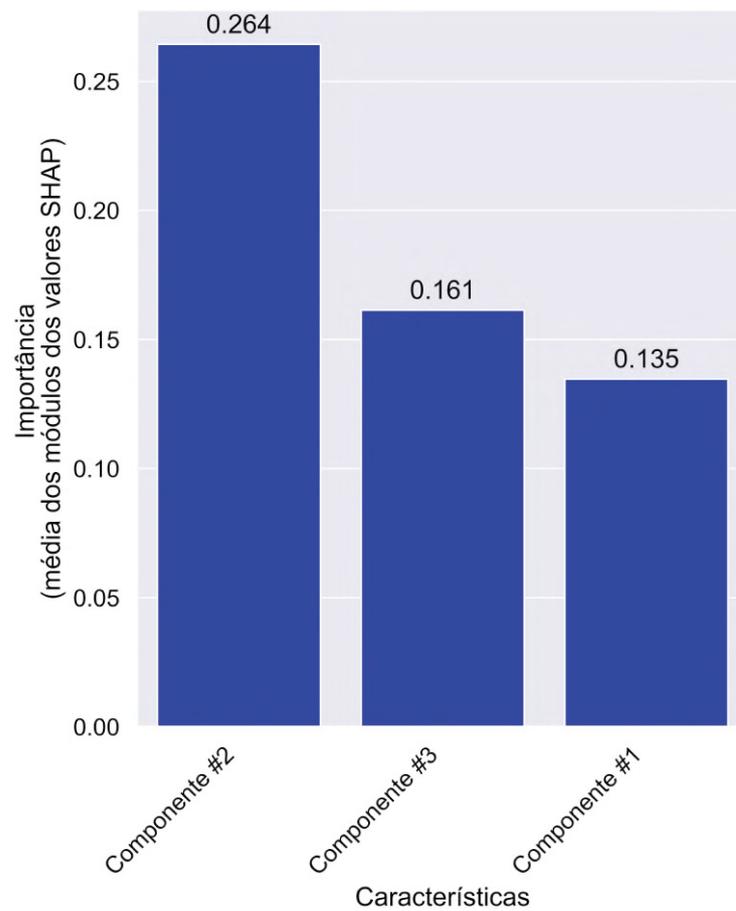
FIGURA 159 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO XGBOOST NO EXPERIMENTO II



FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo XgBoost no experimento II.

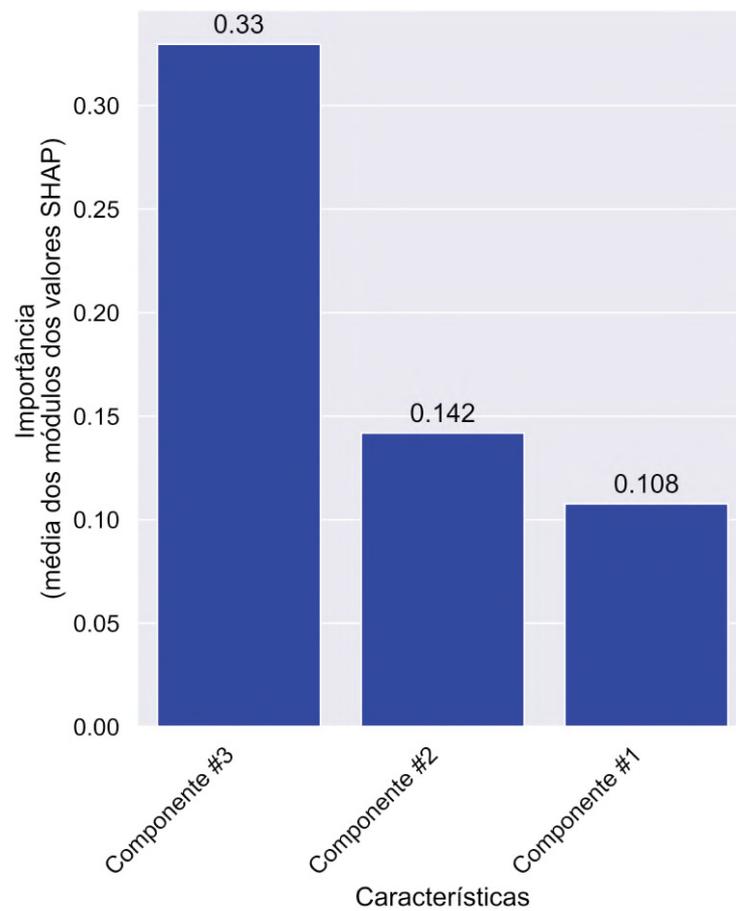
FIGURA 160 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO XGBOOST NO EXPERIMENTO III



FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo XgBoost no experimento III.

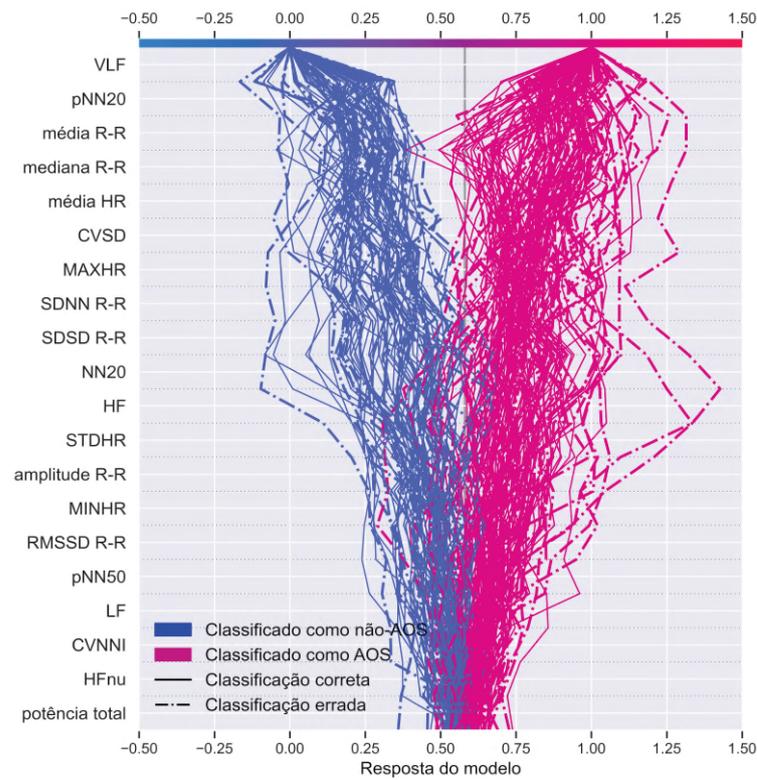
FIGURA 161 – IMPORTÂNCIA GERAL DE CADA CARACTERÍSTICAS PARA O MODELO XGBOOST NO EXPERIMENTO IV



FONTE: Próprio autor

LEGENDA: Importância média, medidos com média dos valores SHAP, de cada características para o modelo XgBoost no experimento IV.

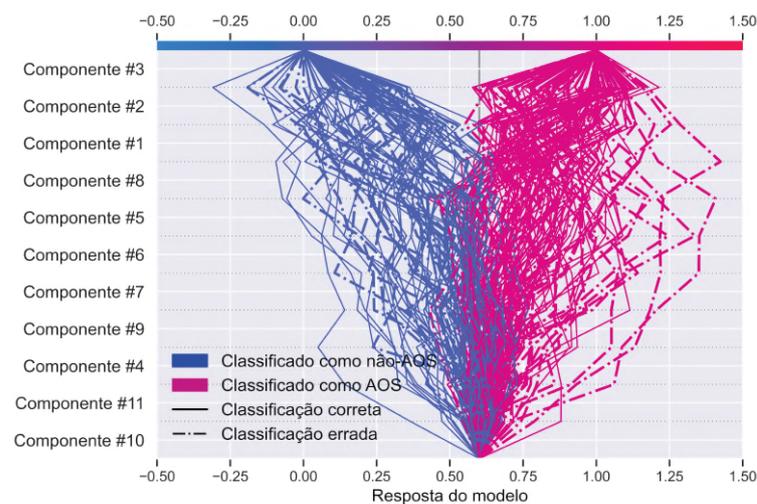
FIGURA 162 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO XGBOOST NO EXPERIMENTO I



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo XgBoost no experimento I.

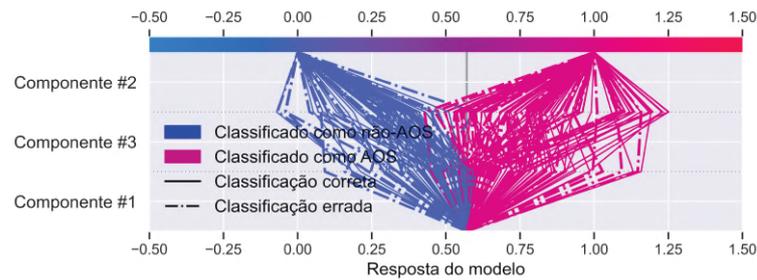
FIGURA 163 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO XGBOOST NO EXPERIMENTO II



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo XgBoost no experimento II.

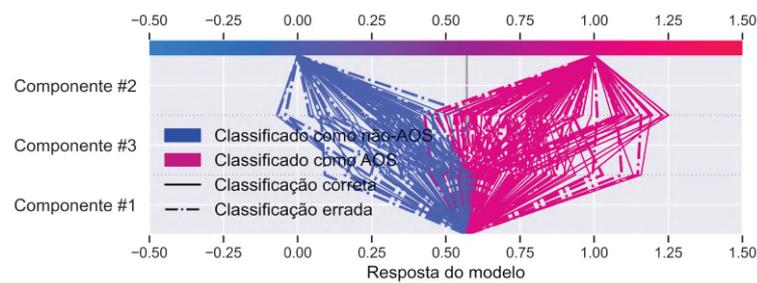
FIGURA 164 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO XGBOOST NO EXPERIMENTO III



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo XgBoost no experimento III.

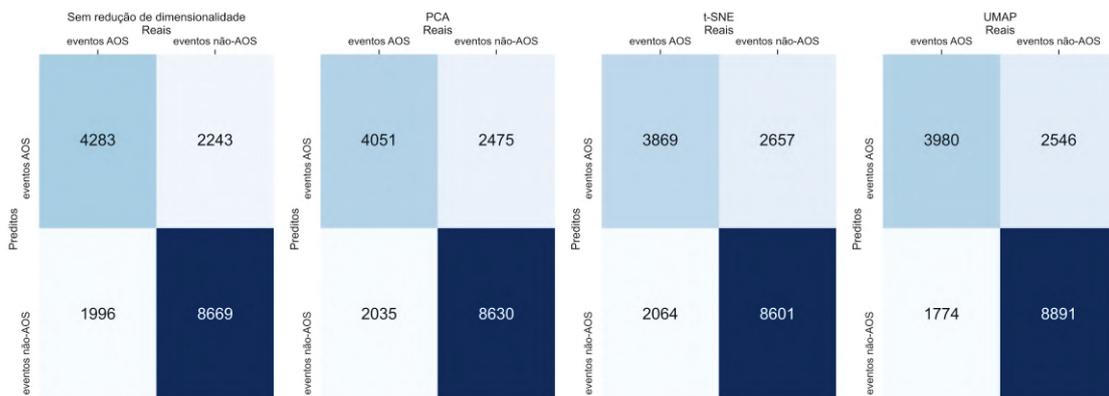
FIGURA 165 – CONTRIBUIÇÃO DE CADA CARACTERÍSTICA PARA INFERÊNCIA DA AMOSTRA DE 200 DE EXEMPLOS DO MODELO XGBOOST NO EXPERIMENTO IV



FONTE: Próprio autor

LEGENDA: Contribuição de cada característica no processo decisório da inferência de toda amostra do modelo XgBoost no experimento IV.

FIGURA 166 – MATRIZ DE CONFUSÃO DO MODELO XGBOOST



FONTE: Próprio autor

LEGENDA: Matriz de confusão para o modelo xgboost em cada um dos experimentos.

TABELA 70 – RESULTADOS DOS TESTES DE HIPÓTESES E TEMPOS DE PREDIÇÃO DA BASE DE TESTE PARA MODELOS ML

Medição	Experimento/Modelo	XgBoost	RF	SVM	NB	AdaBoost	VM	MLP
Tempo de predição (s)		0,39	1,72	14,53	6,94 × 10 <sup>-3</sup>	1,60	116,50	9,43 × 10 <sup>-3</sup>
p-Valor do teste binomial	Sem redução	2,43 × 10 <sup>-60</sup>	8,48 × 10 <sup>-65</sup>	6,38 × 10 <sup>-69</sup>	4,32 × 10 <sup>-22</sup>	5,62 × 10 <sup>-70</sup>	8,33 × 10 <sup>-62</sup>	3,77 × 10 <sup>-55</sup>
p-Valor do teste McNema		1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>
Tempo de predição (s)		0,40	2,19	12,95	4,49 × 10 <sup>-3</sup>	0,59	73,73	9,92 × 10 <sup>-3</sup>
p-Valor do teste binomial	PCA	7,01 × 10 <sup>-53</sup>	7,39 × 10 <sup>-60</sup>	1,90 × 10 <sup>-69</sup>	3,77 × 10 <sup>-38</sup>	2,64 × 10 <sup>-65</sup>	7,01 × 10 <sup>-53</sup>	4,50 × 10 <sup>-56</sup>
p-Valor do teste McNema		1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>
Tempo de predição (s)		0,11	1,60	13,24	1,98 × 10 <sup>-3</sup>	0,07	126,44	0,03
p-Valor do teste binomial	t-SNE	1,16 × 10 <sup>-47</sup>	2,29 × 10 <sup>-49</sup>	3,77 × 10 <sup>-55</sup>	6,17 × 10 <sup>-49</sup>	7,01 × 10 <sup>-53</sup>	1,08 × 10 <sup>-54</sup>	8,83 × 10 <sup>-38</sup>
p-Valor do teste McNema		1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>
Tempo de predição (s)		0,01	0,93	14,35	1,49 × 10 <sup>-3</sup>	0,73	130,45	0,02
p-Valor do teste binomial	UMAP	6,02 × 10 <sup>-58</sup>	4,39 × 10 <sup>-48</sup>	6,74 × 10 <sup>-59</sup>	4,50 × 10 <sup>-56</sup>	3,10 × 10 <sup>-54</sup>	2,49 × 10 <sup>-53</sup>	5,26 × 10 <sup>-57</sup>
p-Valor do teste McNema		1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>

FONTE: Elaborada pelo autor

TABELA 71 – RESULTADOS DOS TESTES DE HIPÓTESES E TEMPOS DE PREDIÇÃO DA BASE DE TESTE PARA MODELOS ML

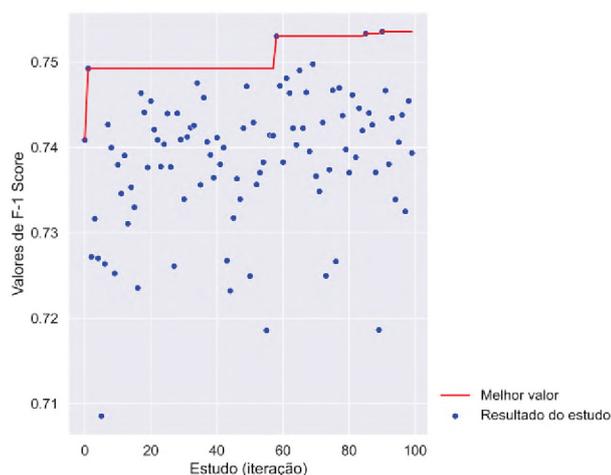
Medição	Experimento/Modelo	LR	LightGBM	Bagging	DT
Tempo de predição (s)		9,93 × 10 <sup>-3</sup>	0,08	2,93	2,97 × 10 <sup>-3</sup>
p-Valor do teste binomial	Sem redução	4,50 × 10 <sup>-56</sup>	2,68 × 10 <sup>-62</sup>	8,62 × 10 <sup>-64</sup>	1,15 × 10 <sup>-50</sup>
p-Valor do teste McNema		1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>
Tempo de predição (s)		4,96 × 10 <sup>-3</sup>	0,22	0,90	1,15 × 10 <sup>-3</sup>
p-Valor do teste binomial	PCA	8,81 × 10 <sup>-54</sup>	5,26 × 10 <sup>-57</sup>	6,02 × 10 <sup>-58</sup>	9,23 × 10 <sup>-45</sup>
p-Valor do teste McNema		1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>
Tempo de predição (s)		9,76 × 10 <sup>-3</sup>	0,04	11,65	1,15 × 10 <sup>-3</sup>
p-Valor do teste binomial	t-SNE	3,77 × 10 <sup>-38</sup>	5,47 × 10 <sup>-52</sup>	4,18 × 10 <sup>-51</sup>	2,84 × 10 <sup>-39</sup>
p-Valor do teste McNema		1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>
Tempo de predição (s)		9,92 × 10 <sup>-3</sup>	0,02	19,98	5,20 × 10 <sup>-3</sup>
p-Valor do teste binomial	UMAP	3,13 × 10 <sup>-50</sup>	3,77 × 10 <sup>-55</sup>	3,13 × 10 <sup>-50</sup>	1,96 × 10 <sup>-52</sup>
p-Valor do teste McNema		1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>	1,52 × 10 <sup>-23</sup>

FONTE: Elaborada pelo autor

## DEMAIS RESULTADOS PARA REDES RECORRENTES DO TIPO *LONG-SHORT TERM MEMORY*, LSTM

Na FIGURA 167 é mostrado o histórico de otimização do modelo de LSTM em função dos valores de F1-score de cada configuração testada na base de validação. As configurações apresentaram resultados entre cerca de 0,71 e de 0,75 de F1-score. Na FIGURA 168 são mostrados a importância relativa de cada hiperparâmetro no resultado e no tempo de otimização. No gráfico da parte superior, "a", que mostra a importância, é possível observar que o número de unidades na camada 1 de LSTM foi o hiperparâmetro mais importante, enquanto que o hiperparâmetro de número de camadas de LSTM foi o que praticamente dominou o tempo de otimização, mostrado no gráfico da parte inferior, "b".

FIGURA 167 – HISTÓRICO DE OTIMIZAÇÃO PARA MODELO LSTM



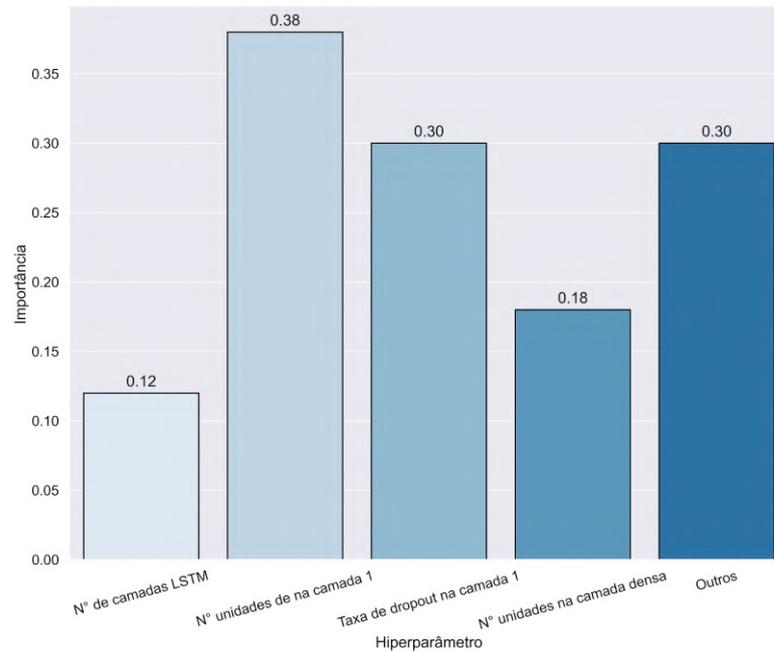
FONTE: Próprio autor

LEGENDA: Histórico de otimização para o modelo de LSTM com destaque para os melhores valores de F1-score obtidos.

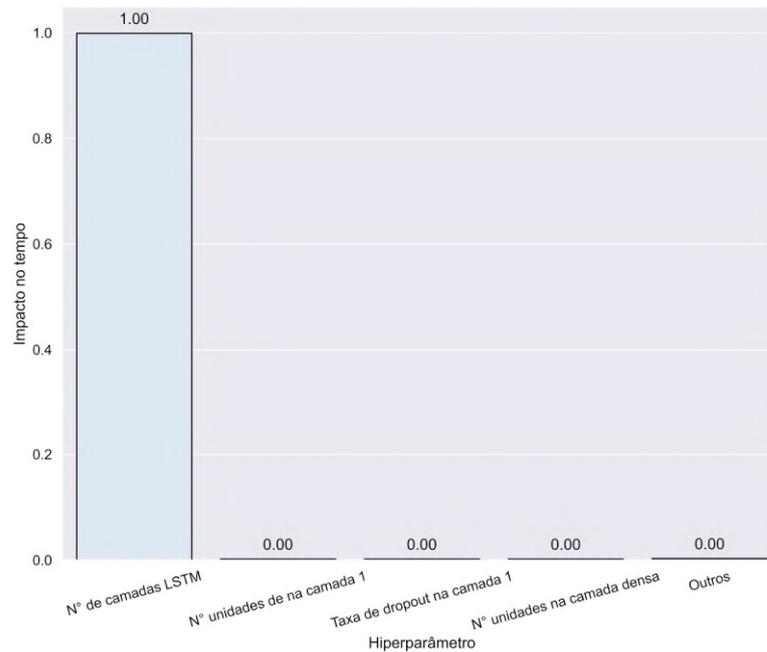
## DEMAIS RESULTADOS PARA MODELO DE DL HÍBRIDO DE REDES CONVOLUCIONAIS E REDES RECORRENTES, CNN E LSTM

Durante a otimização do modelo híbrido, as configurações testadas obtiveram valores de F1-score entre cerca de 0,78 e 0,81, com duas configurações com valores abaixo, com aproximadamente 0,72 e 0,60, conforme mostrado na FIGURA 169. Quando avaliados o impacto relativo no resultado, mostrado no gráfico da parte superior "a" da FIGURA 170, houve uma distribuição, com o hiperparâmetro que controla a aplicação ou não de uma camada de normalização da saída da CNN como que mais contribuiu para os resultados durante a otimização. O tempo de otimização foi ditado praticamente pelo hiperparâmetro que controla o número de camadas de LSTM da rede, mostrado no gráfico "b", da parte inferior, da mesma FIGURA. Como a quantidade

FIGURA 168 – IMPORTÂNCIA RELATIVA NO RESULTADO E IMPACTO RELATIVO NO TEMPO DOS HIPERPARÂMETROS DURANTE OTIMIZAÇÃO DO MODELO LSTM



(a) importância relativa no desempenho



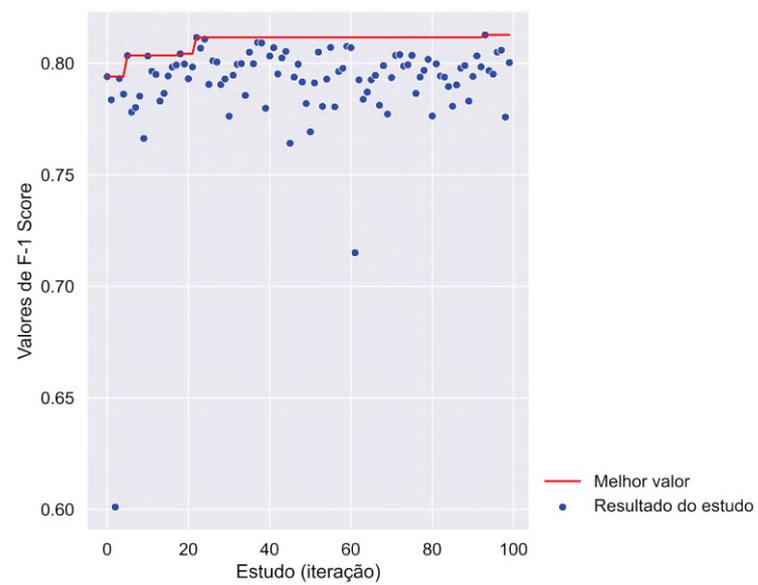
(b) impacto relativo no tempo de otimização

FONTE: Próprio autor

LEGENDA: Impacto no resultado e no tempo de otimização de cada hiperparâmetro durante otimização do modelo de LSTM.

de camadas de CNN foram fixas, adicionar mais ou menos camadas de LSTM foi que mais impactou o tempo total de otimização.

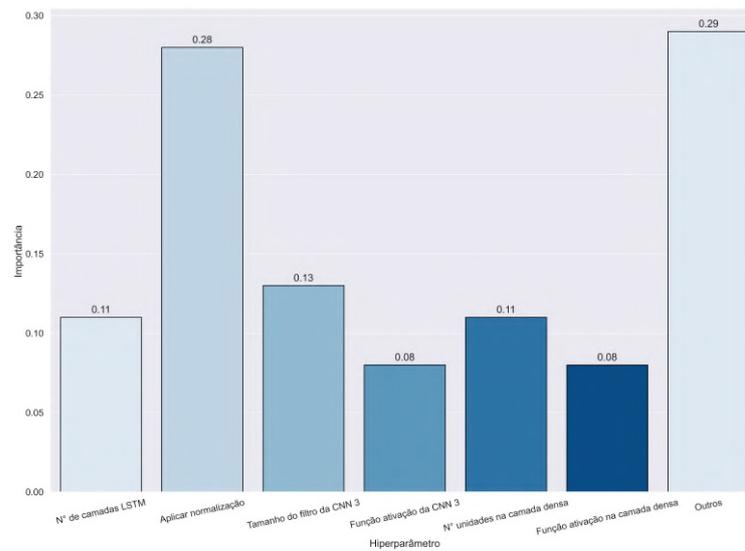
FIGURA 169 – HISTÓRICO DE OTIMIZAÇÃO PARA MODELO HÍBRIDO



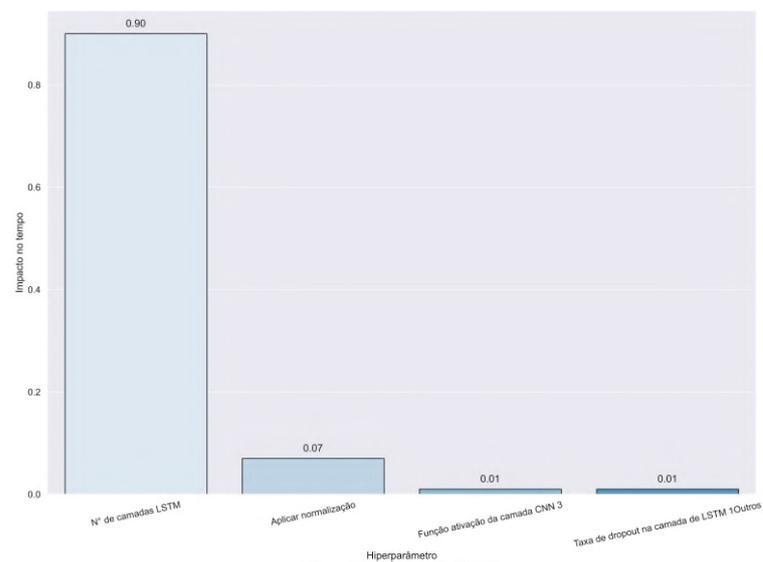
FONTE: Próprio autor

LEGENDA: Histórico de otimização para o modelo de híbrido com destaque para os melhores valores de F1-score obtidos.

FIGURA 170 – IMPORTÂNCIA RELATIVA NO RESULTADO E IMPACTO RELATIVO NO TEMPO DOS HIPERPARÂMETROS DURANTE OTIMIZAÇÃO DO MODELO HÍBRIDO



(a) importância relativa no desempenho



(b) impacto relativo no tempo de otimização

FONTE: Próprio autor

LEGENDA: Impacto no resultado e no tempo de otimização de cada hiperparâmetro durante otimização do modelo de híbrido.